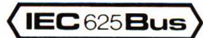


IEC-Bus

Grundlagen – Technik – Anwendungen

IEC-Bus-Peripheriegeräte



PPC



Process Controller PPC für IEC-Bus-Meßsysteme

Wirtschaftliches IEC-Bus-Steuergerät, aber auch preisgünstiger Tischrechner mit Programmierkomfort.

- Alle IEC-Bus-Befehle im Klartext
- Alle IEC-Bus-Optionen
- Treiber-Softwarepaket für alle R&S-Geräte
- Floppy Disk (80 kByte) eingebaut, zwei externe Floppy-Stationen möglich
- Programmierung in Basic
- Störarmes Netzteil, Netzfilter
- Option: RS-232-C-Schnittstelle

Dazu: **Universal-Drucker PUD** (ohne Bild) 45 Zeichen/s · 80/132 Zeichen/Zeile · Papiertransport: ½ und 1 Zeile · 7 × 9 oder 11 × 9 Matrix-Druckkopf · Schnittstelle: 8 bit parallel · Option: IEC-Bus

Relais-Matrix PSN (DC und NF)
Sechs Reed-Schnellrelais, zwei Leistungsrelais. NF- und Steueranwendungen, hohe Belastbarkeit. Leuchtdiodenanzeige.



PSN



PTC

Digitalthermometer PTM −100 ... + 300 °C mit Pt 100-Meßfühlern (DIN 43 760) in Vierleiterschaltung, Temperaturdifferenzmessung.
Temperaturregler PTC Daten wie PTM. Zusätzlich: Solltemperatureingabe über Tastenfeld, Schaltzustandsanzeige, Schaltschütz 10 A/220 V und Steuerrelais.

Relais-Matrix PSU
DC ... 6 GHz.
Sechs unabhängige 50-Ω-Koaxialrelais.
HF- und Impulsanwendungen.
Leuchtdiodenanzeige.



PSU



DPSP

HF-Eichleitung DPSP
0...2700 MHz / 0...139 dB
Kleinste Dämpfungsschritte 1 dB, geringe Grunddämpfung, hohe Genauigkeit, kleiner Dämpfungsfehler.
Extrem lange Lebensdauer.
Ohne IEC-Bus: Typ DPS.

IEC-Bus-Meßgeräte

Bereits seit 1974 sind systemfähige R&S-Meßgeräte mit IEC-Bus ausgerüstet z. B.:
Meßempfänger ESH 3, MSUP:
10 kHz bis 1 GHz
Sprechfunkmeßplätze SMPU, SMFP:
bis 1 GHz
Synthesizer SMS, XPC: 50 kHz bis 1,36 GHz
Vector-Voltmeter ZPV, ZPV-Z 5:
0,1 MHz bis 2 GHz
HF-, DC-, NF-Voltmeter URV 4, UDS 6:
bis 2 GHz
Modulationsmeter FAM: 50 kHz bis 1,36 GHz
Rauschgenerator SUF 2: 50 MHz
Dazu die Strom/ Spannungsquellen NGPS, NGPU.
Weitere IEC-Bus-Geräte sind in Vorbereitung.

Berlin · Ruf (0 30) 3 41 40 36
Hamburg · Ruf (0 40) 3 80 19-1
Köln · Ruf (02 21) 77 22-1
Neu-Isenburg · Ruf (0 61 02) 31 36
Karlsruhe · Ruf (07 21) 2 79 81
München · Ruf (0 89) 41 62-1
Nürnberg · Ruf (09 11) 6 48 81



**Interkama
Düsseldorf
Halle 1, Stand 1 A 01**

Der IEC-Bus gestern – heute – morgen

Auf dem Tisch stehen mehrere elektronische Meßgeräte. Ein Ingenieur sitzt mit Bleistift und Papier davor und führt Meßreihen durch. Irgendwann wird ihm diese Tätigkeit zu langweilig. Er steht auf, holt einen Tischrechner aus dem Regal, stellt ihn neben die Meßgeräte und verbindet alles über Standardkabel miteinander. Nach einer Stunde Programmierarbeit erhebt er sich vergnügt zu einer Erholungspause, währenddessen der Rechner fleißig die ihm aufgetragenen Meßaufgaben durchführt und auf dem Drucker ausgibt.

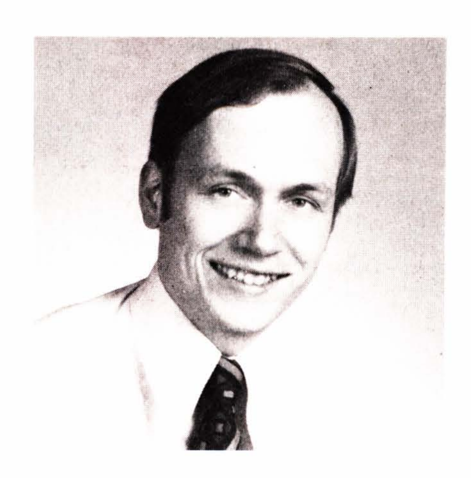
Im Bereich der elektronischen Meßtechnik ist dieses inzwischen ein gewohntes Bild geworden, doch war die Entwicklung dahin keineswegs selbstverständlich und viele Dinge mußten zusammenkommen, um das heutige Ergebnis zu ermöglichen:

Als man die Notwendigkeit einer einheitlichen Datenschnittstelle für Meßgeräte erkannte, wurde im Rahmen der IEC eine internationale Arbeitsgruppe gegründet, die die Aufgabe hatte, eine gemeinsame, von allen Ländern akzeptierte Norm zu erarbeiten. Verschiedene Länder- und Firmenvorschläge wurden diskutiert. Entscheidend war, daß man auf die sonst üblichen nationalen Egoismen verzichtete und sich auf den Vorschlag eines großen Meßgeräteherstellers als Arbeitsgrundlage einigte. Aufgrund dessen wurden bereits 1975 die ersten Versuche gemacht, IEC-Bus-Geräte auch verschiedener Hersteller zusammenzuschalten – die ELEKTRONIK berichtete damals schon darüber.

Den nächsten wichtigen Impuls gab dem IEC-Bus der inzwischen weit verbreitete Mikroprozessor. Erst durch ihn wurde es möglich, preiswerte Meßgeräte mit IEC-Bus zu realisieren.

Gleichzeitig kamen dann auch IEC-Bus-kompatible Rechner mit relativ niedrigen Preisen auf den Markt. Den nächsten entscheidenden Schritt bewirkte die Halbleiterindustrie, die die IEC-Bus-Funktionen in einem Chip integrierte. Nun war der IEC-Bus nicht mehr aufzuhalten, und er wurde zum unbeschränkten Herrscher auf dem Gebiet der universellen Bit-parallelen Interface-Systeme. Die Anwendungen sind inzwischen so vielfältig geworden, daß es fast nicht mehr möglich ist, eine genaue Zahl der Geräte mit IEC-Bus-Anschluß zu nennen. Für den in diesem Maße kaum vorhersehbaren Erfolg trafen also zahlreiche Voraussetzungen zusammen:

Das rechtzeitige Erkennen der Standardisierungsnotwendigkeit, bevor



Einzelentwicklungen vollendete Tatsachen geschaffen hatten, das Vorhandensein der richtigen Technologie, die wirtschaftliche Notwendigkeit der Problemlösung, das Zurückstellen von nationalen und Firmeninteressen und sicher nicht zuletzt der Einsatz einer kompetenten und engagierten Normengruppe.

Ist damit nun der Höhepunkt der IEC-Bus-Technik erreicht? Sicher nicht. Sowohl auf dem Gebiet der Tischrechner, als auch im Bereich der IEC-Bus-Meßgeräte der unteren Preisklasse ist z. Z. ein weiteres Absinken im Preisniveau zu verzeichnen. Ein vollständiger rechnergesteuerter IEC-Bus-Automat in der Preisklasse unter 10 000.– DM ist durchaus keine Utopie mehr. Damit ist die preisliche Voraussetzung gegeben, um der IEC-Bus-Technik zu einer echten Massenanwendung zu verhelfen.

Darüber hinaus deutet sich aber eine weitere Entwicklung an. Nachdem es bisher trotz langjähriger Versuche nicht gelungen ist, eine Schnittstelle für Computerperipherie zu standardisieren, bietet sich hier der IEC-Bus als Ausweg an. Obwohl ursprünglich nicht für diese Anwendungen konzipiert, ist er z. Z. die einzig mögliche weltweit standardisierte Datenschnittstelle zum Anschluß von Druckern, Lesern, Magnetbandkassetten und Floppylaufwerken. Insbesondere im Tischrechnerbereich wird sich deshalb der IEC-Bus sehr schnell durchsetzen, nachdem führende Hersteller von Tischrechnern hier bereits Maßstäbe gesetzt haben. Entsprechend der Größe dieses Marktes existiert hier ein noch weit größeres Potential von IEC-Bus-Anwendungen als im Bereich der elektronischen Meßtechnik. Die Dynamik dieser Entwicklung ist so groß, daß sich inzwischen auch die entsprechenden Normgremien mit einer Anlehnung an den IEC-Bus beschäftigen.

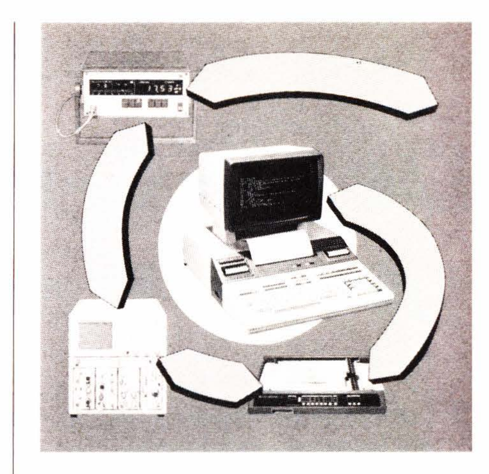
Im Bereich Bit-parallele Datenverbindungen über kurze Entfernungen wird deshalb der IEC-Bus seinen Siegeszug weiter fortsetzen und noch viele Jahre wichtigster Industriestandard bleiben.

Und was kommt danach?

Der IEC-Bus ist nicht in der Lage, Datennetze innerhalb eines Gebäudekomplexes oder einer Fabrik mit größeren Teilnehmerzahlen zu realisieren. Hier besteht die Notwendigkeit für einen Bit-seriellen Bus hoher Übertragungsrate, der Koaxialleitungen oder Lichtleiter als Übertragungsmedium verwendet. Dieser serielle Bus ist eine Notwendigkeit im Bereich der lokalen Datennetze. Er wird deshalb in kurzer Zeit Realität sein und neben dem IEC-Bus ein sinnvolles Dasein führen. Beide Systeme werden sich ergänzen, wobei auch Kombinationen möglich sind.

Infolgedessen wird auch in Zukunft niemand um Theorie und Praxis des IEC-Bus herumkommen, und es ist sicherlich ein günstiger Zeitpunkt, wenn die ELEKTRONIK in einer Sonderausgabe die zahlreichen Informationen zusammenstellt, die sie in den letzten Jahren über den IEC-Bus veröffentlicht hat, ergänzt aber auch durch vollkommen neues Material, wie z. B. über die Bus-Interface-ICs. *Tonio Frühauf*

Inhalt



Vorwort

Grundlagen

Technik

Normung

Anwendungen

Kurzberichte

IEC-Bus

Scheinbar einfach, wie es auch das Titelbild veranschaulicht, verbindet der IEC-Bus die verschiedensten Geräte miteinander zu einem automatischen Meßsystem. Doch ganz so einfach geht es in der Praxis nicht: Auch nach langen Jahren der Normungsarbeit in nationalen und internationalen Gremien kommt der Anwender nicht darum herum, mit fundierten Eigenkenntnissen an die Planung und den Betrieb eines automatischen Meßsystems heranzugehen. „Fundiert" bedeutet in diesem Zusammenhang vor allem „festsitzende Grundlagen", ergänzt durch praxisnahe Erfahrungen.
Und genau hier soll unser Sonderheft eine Unterstützung sein. Es bringt in einem breiten Spektrum, beginnend bei den IEC-Bus-Grundfunktionen und endend mit neuesten Anwendungsberichten aus der Industrie, einen umfassenden Überblick über den momentanen Stand der IEC-Bus-Technik.

Die Redaktion

1980
Franzis-Verlag GmbH, Karlstr. 37, 8000 München 2.
Produktion: ELVAG, Elektronik Verlag Luzern AG, CH-6002 Luzern.
Sonderheft der Zeitschrift ELEKTRONIK.
Redaktion: Wolfgang Hascher, für den Text verantwortlich: Günther Klasche.

ISSN 0170-0898
Druck: Stämpfli, CH-Bern
ZV-Artikel-Nr. 47021 · St/ZV/880/328/6'

Immer häufiger werden elektronische Meßgeräte nicht mehr einzeln benutzt, sondern in Systeme eingegliedert und damit teilweise oder ganz automatisiert. Eine international genormte ,,Sammelschiene'', die das ermöglicht, ist der IEC-Bus. Doch zu seinem Verständnis sind Grundlagen äußerst wichtig – dieser Beitrag schließt hier eventuelle Wissenslücken, wobei die vom Autor zweckmäßigerweise gewählte ,,Wir''-Form aus Gründen der besseren Verständlichkeit beibehalten wurde.

Dipl.-Ing. Jürgen Klaus

Wie funktioniert der IEC-Bus?

1. Teil

Zusammenschluß von Geräten

Sender und Empfänger

Das kleinste denkbare Meßsystem besteht aus zwei Geräten. *Bild 1* zeigt ein Beispiel für einen solchen Zusammenschluß: Ein Sender „S" gibt über einen parallelen Ausgang Daten ab, die von einem Empfänger „E" empfangen werden. Ein Beispiel hierfür ist ein Voltmeter mit einem angeschlossenen Drucker. Die Daten werden parallel in BCD-Form transportiert. Nach jeder Messung erfolgt eine Datenmeldung, z. B. ein Druckbefehl. Der Empfänger quittiert die Daten mit einer Fertigmeldung.

Um den Leser an die Namen und Bezeichnungen aus der IEC-Bus-Norm zu gewöhnen, werden in allen Bildern und Texten neben deutschen Ausdrücken auch noch die entsprechenden Benennungen in englischer Sprache benutzt, wie sie später in der IEC-Bus-Norm verwendet werden.

Der Druckbefehl aus unserem ersten Beispiel würde in der IEC-Bus-Norm „*Data valid*" heißen: „DAV". Die Quittung vom Empfänger heißt dann *„data accepted"* und wird mit „DAC" abgekürzt.

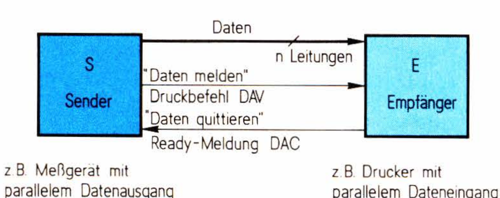
Solange man wirklich nur einen Sender mit einem Empfänger verbinden will, ist es relativ einfach. Hat man aber mehrere Sender, z. B. mehrere Meßgeräte, und mehrere Empfänger, d. h. mehrere Geräte, die in irgendeiner Form Daten aufschreiben, so ist mit einer einfachen Verbindung nach Bild 1 nicht mehr viel zu machen.

Sternsysteme

Von verschiedenen Herstellern wurden und werden auch noch Sternsysteme geliefert; ein solches besteht aus einem zentralen Gerät und mehreren sternförmig angeschlossenen anderen Geräten. Ein prinzipielles Beispiel hierfür ist in *Bild 2* dargestellt. In der Mitte des Bildes befindet sich ein Koppler, in den mehrere Eingangskarten $I_1 ... I_n$ und mehrere Ausgangskarten $O_1 ... O_m$ eingesteckt werden können.

Da man normalerweise nicht davon ausgehen kann, daß die Sender „S" alle mit der gleichen Ausgangsschnittstelle versehen sind, gehört zu jedem Sender eine spezielle Eingangskarte „I" mit einem dazu passenden Verbindungskabel „K". Entsprechend sieht es auf der Ausgabeseite aus. Die Empfänger in einem solchen System sind z. B. Drucker, Fernschreiber oder Lochstreifenstanzer, d. h. also Geräte, die Daten ausgeben können. Leider haben auch diese Geräte verschiedene Schnittstellen. Deshalb gehört zu jedem Empfänger „E" eine besondere Ausgangskarte „O" und ein besonderes Anschlußkabel „A". Der Koppler enthält noch zusätzliche Elemente, die zur Steuerung, Formatierung und Codeumsetzung notwendig sind.

In einem solchen Sternsystem lassen sich die Daten von verschiedenen Meßgeräten auf verschiedene Ausgabegeräte übertragen. Im Steuerteil des Kopplers sind meist einige Funktionen einstellbar, wie z. B. das Drucken von Zwischenräumen zwischen den einzelnen Meßwerten. Eines der Geräte auf der Senderseite könnte z. B. noch eine Digitaluhr sein, so daß beim Aufschreiben der Daten auch das Erfassen von Zeiten möglich ist. Diese Uhr kann den Koppler zu bestimmten Zeiten starten.



◀ Bild 1. Einfache parallele Verbindung von zwei Geräten

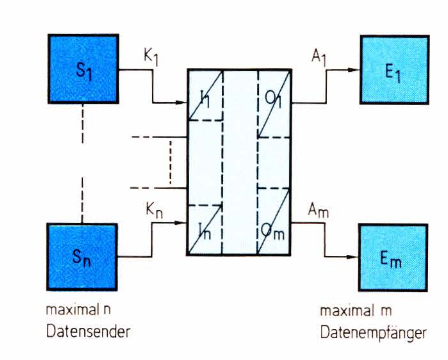
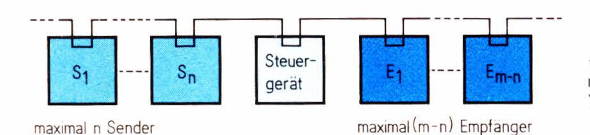


Bild 2. ▶ Sternsystem mit Koppler



◀ Bild 3. Bus-System mit „durchgeschleifter" Verbindung der Geräte

Ein solches Sternsystem hat gewisse Vorteile für einfache Anwendungen, da der Verwender nur die Karten zu kaufen braucht, die in seinem speziellen System erforderlich sind. Andererseits ist man mit dem Kauf des Kopplers auf die Verwendung von Meß- und Ausgabegeräten eines einzigen Herstellers festgelegt, da normalerweise nicht für jedes beliebige auf dem Markt befindliche Gerät Eingangs- oder Ausgangskarten zur Verfügung stehen, die in den speziellen Koppler hineinpassen. Die Steuerung des zeitlichen Ablaufs der einzelnen Funktionen ist nur auf sehr einfache Art und Weise möglich. *Rechenoperationen, Vergleichsoperationen oder andere komplizierte Aufgaben sind in einem solchen System überhaupt nicht möglich.*

Außerdem muß der Koppler für verschieden schnelle Empfangsgeräte, die gleichzeitig arbeiten sollen, mit einer Überwachung der Empfangsgeräte ausgerüstet sein. Will man in einem solchen System einen Fernschreiber einsetzen, der mit dem CCITT-Code arbeitet, so benötigt man dafür auf jeden Fall einen Codeumsetzer, da der CCITT-Code nichts mit den üblicherweise parallel einkommenden digitalen Daten zu tun hat. Das bedeutet, daß ein Fernschreiber, wie er in Deutschland oft angetroffen wird, nur über einen relativ teuren Codeumsetzer in einem solchen System verwendet werden kann. Ein solches Sternsystem ist deshalb nicht besonders flexibel und nur für einfache Anlagen zu empfehlen. Die Erweiterbarkeit auf eine größere Zahl von Geräten ist meist begrenzt durch die Zahl der Plätze für Eingangs- und Ausgangskarten im Koppler.

Externes Bus-System

Der nächste Schritt auf dem Weg zum IEC-Bus-System ist ein externes Bus-System (*Bild 3*). In einem Sternsystem, wie es z. B. in Bild 2 skizziert wurde, gibt es innerhalb des Kopplers meistens eine Art von Bus-Verbindungs-Leitung; in diese werden die Eingangs- und die Ausgangskarten eingesteckt. Von der Steuereinheit des Kopplers werden die Eingangs- und Ausgangskarten dann gesteuert.

Im Gegensatz dazu liegt das Bus-System nach Bild 3 *außerhalb* der Geräte. Hierbei werden nicht spezielle Kopplerkarten passend zu verschiedenen Geräten in einem Kopplergerät betrieben, sondern alle Geräte, die in einem solchen Bus-System arbeiten sollen, werden mit einer definierten gleichen *Schnittstelle* versehen. Die Verbindung erfolgt mit Standardkabeln auf der Rückseite der Geräte, so daß eine durchgehende Verbindung über alle Geräte geschaffen wird. Innerhalb der Geräte wird dann eine Verbindung zwischen dem eigentlichen Gerät und dem Standardanschluß hergestellt. Ein Interface, wie es für ein solches externes Bus-System notwendig ist, wird später noch näher erläutert.

Bus-Systeme mit programmierbarem Steuergerät

Da in einem Bus-System *alle* Leitungen an *alle* Geräte gehen, muß jetzt innerhalb der Geräte etwas vorgesehen werden, um den angeschlossenen Geräten die Feststellung zu ermöglichen, ob die Daten, die über den Bus verschickt werden, für das Gerät bestimmt sind oder nicht. Ein Bus-System ist ohne die *Adressierung* der einzelnen Geräte nicht möglich (*Bild 4*).

In einem Bus-System dieser Art muß immer ein *Steuergerät* vorhanden sein, welches die einzelnen Geräte nacheinander aufruft; es muß wissen, welches Gerät als nächstes zu einer Aktion veranlaßt werden soll und wie diese Aktion veranlaßt

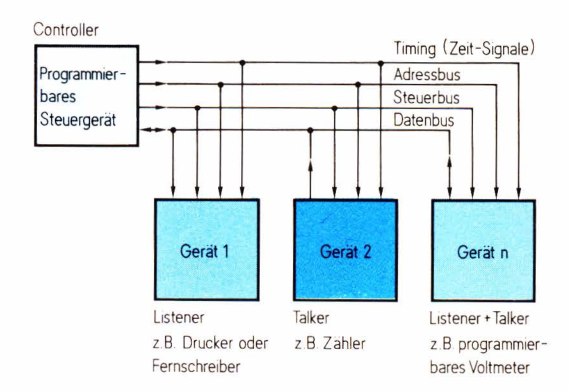


Bild 4. Schema eines Bus-Systems mit mehreren Busleitungen

wird. Die dazu notwendigen Informationen befinden sich innerhalb des *Programms* dieses Steuergerätes; es muß veränderbar sein, um ein solches System flexibel genug zu halten. Darum wird ein solches Steuergerät üblicherweise programmierbar ausgeführt; im IEC-Bus heißt ein entsprechendes Gerät *Controller*.

Das programmierbare Steuergerät arbeitet schrittweise sein Programm ab; wenn darin steht, daß ein bestimmtes Gerät aufgerufen und zu einer bestimmten Aktion veranlaßt werden soll, so schickt das Steuergerät als erstes die Adresse dieses ausgewählten Gerätes über den Adreßbus. Jedes Gerät hat eine bestimmte *Geräteadresse,* und wenn diese auf dem Adreßbus erscheint, so achtet das Gerät auf die nachfolgenden Steuerbefehle oder Daten. Wenn z. B. auf Bild 4 das Gerät „n" auf einen bestimmten Meßbereich programmiert werden soll, so muß das Steuergerät als erstes die Geräteadresse dieses Voltmeters aussenden. Anschließend wird über den Steuerbus ein *Steuerbefehl* gegeben, der dem Voltmeter sagt, daß die auf dem Datenbus gleichzeitig ausgeschickten Daten ein Signal für den Meßbereich sind. Nach der Übernahme dieser Information ist das Voltmeter programmiert. Die Übernahme selbst wird gesteuert durch Zeit-Signale *(Timing)*, wie z. B. die Takt-Signale *(clock pulse).*

Wenn das Voltmeter anschließend eine Messung ausführen soll, so muß als erstes wieder der Adreßbus die Adresse des Voltmeters enthalten, und anschließend über den Steuerbus der Fernsteuerbefehl für den Beginn einer Messung gesendet werden. Für einen solchen Steuerbefehl ist es nicht erforderlich, über den Datenbus Informationen zu verschicken. Im nächsten Schritt müßte das Steuergerät die Daten aus dem Voltmeter abfragen. Durch zusätzliche Zeitsignale muß dafür gesorgt werden, daß nicht eher Daten aus dem Voltmeter abgefragt werden, als bis es auch wirklich die verlangte Messung ausgeführt hat. Erst wenn das Signal für die Beendigung der Messung erfolgt ist, darf aus dem Voltmeter wirklich der Meßwert abgerufen werden.

Das Steuergerät gibt dazu über den Adreßbus wieder die Adresse des Voltmeters aus und über den Steuerbus mehrere Befehle, um nacheinander die einzelnen Ziffern des Meßwertes über den Datenbus abzurufen; sie gelangen über den Datenbus zum Steuergerät und werden dort zwischengespeichert.

Um den so erhaltenen Meßwert ausdrucken zu können, gibt das Steuergerät die Adresse des Druckers aus, und anschließend werden Signale an den Drucker gegeben, damit er weiß, daß die gleichzeitig auf dem Datenbus ankommenden Datensignale für ihn bestimmt sind und daß er sie drucken oder zwischenspeichern soll. Bei einem Zeilendrucker gibt es einen weiteren Steuerbefehl, der den Ausdruck des kompletten Meßwertes veranlaßt, sobald dieser im Gerät angekommen ist.

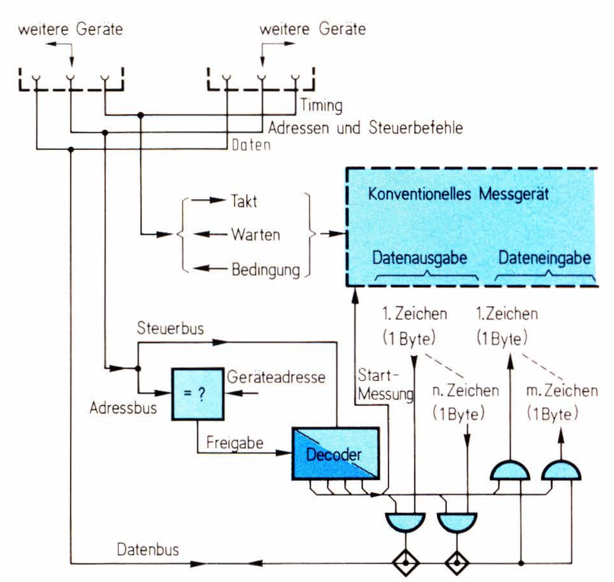


Bild 5. Interface für ein Bus-System mit mehreren Busleitungen

Das Interface: notwendiger Bestandteil universeller Bus-Systeme

Das Voltmeter und der Drucker müssen für eine solche Anwendung mit einem Interface versehen werden, das die Signale für den Bus empfängt bzw. abgibt. Das bedeutet, daß normalerweise kein Gerät konventioneller Bauart mit der richtigen Schnittstelle versehen ist, daß aber alle Geräte sich mit einer solchen bauen lassen würden. Die Vorteile eines solchen Systems sind folgende: Es gibt keine prinzipielle Grenze für die Zahl der anschließbaren Geräte. Jedes Gerät ist über eine relativ einfache Anschlußeinheit an das Bus-System anschließbar. Die Zahl der in einem solchen System anschließbaren Geräte ist in der Praxis aus Belastbarkeitsgründen zwar nicht beliebig groß, aber sie kann doch recht groß gehalten werden.

In einem solchen Bus-System lassen sich normale Standardgeräte verwenden, was bedeutet, daß man relativ billige Geräte im System verwenden kann. Das einzige spezielle Gerät ist das programmierbare Steuergerät, und von ihm hängt es ab, wie flexibel das Gesamtsystem ist. Als Maximallösung für ein solches Steuergerät ließe sich z. B. ein *Minicomputer* verwenden, der ein Ändern des Programms ermöglicht und der außerdem jede Möglichkeit für die Weiterverarbeitung der anfallenden Meßwerte bietet. Für Anwendungen, wo ein Computer zu groß und zu teuer ist, kann man mit einem kleinen programmierbaren Steuergerät oder durch einen Mikroprozessor doch recht leicht programmierbare Ablaufsteuerungen für ein solches Meßsystem aufbauen.

An *Bild 5* soll kurz erläutert werden, wie ein solches Bus-Interface aussieht. Links oben befinden sich die Doppelbuchsen für den Anschluß von weiteren Geräten über die Standardbusleitung. Rechts in der Mitte des Bildes befindet sich das konventionelle Meßgerät; es hat eine Datenausgabe und eine Dateneingabe. Letztere ist nur dann vorhanden, wenn z. B. das Meßgerät in seinen Meßbereichen programmiert werden kann. Aus dem Timing-Bus gelangen Taktsignale in das Interface. Aus dem Meßgerät selber lassen sich Wartezeiten auf den Timing-Bus ausgeben, um die Meßzeit des Meßgerätes zu überwachen. Die auf dem Adreßbus anliegende Adresse wird in einem Vergleicher mit der einstellbaren Geräteadresse verglichen; erst wenn beide übereinstimmen, wird ein Freigabesignal an den Decoder gegeben. Dieser gibt nur dann die Signale vom Steuerbus weiter, wenn das Freigabesignal vorliegt. Im

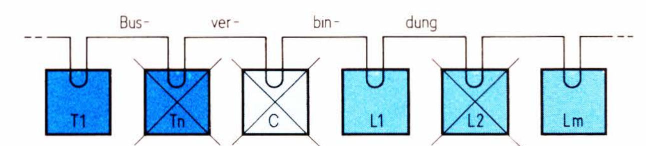


Bild 6. Bus-System mit einem Controller C und mehreren Talkern bzw. Listenern

Decoder werden jetzt die einzelnen Befehle vom Steuerbus decodiert. Mit den einzelnen Codes auf dem Steuerbus werden dann nacheinander die einzelnen Zeichen vom Datenausgang des Meßgerätes auf den Datenbus gegeben, bzw. beim Programmieren des Meßgerätes werden die einzelnen Zeichen, die auf dem Datenbus nacheinander ankommen, an die verschiedenen Stellen gegeben, die bei dem Meßgerät zur Programmierung vorgesehen sind. Ein weiterer Code auf dem Steuerbus dient zum Start des Meßgerätes für eine Messung.

Vorteile einer internationalen Schnittstellennorm

Als Fortentwicklung liegt jetzt eine Schnittstellennorm im Entwurf vor, die gegenüber allem bisher beschriebenen Vorteile hat. Der größte Vorteil für den Anwender liegt darin, daß es sich um eine *internationale Norm* handelt, an die sich immer mehr Hersteller halten werden. Das bedeutet, daß in Zukunft Geräte von verschiedenen Herstellern in einem System kombiniert werden können, ohne daß man bei jedem Hersteller eine andere Ausgangsschnittstelle vorfinden wird.

Bevor auf diese IEC-Bus-Schnittstelle im einzelnen eingegangen wird, folgt jetzt als erstes ein Abschnitt über die Elemente eines Bus-Systems. *Leser, die bereits selbst mit Bus-Systemen gearbeitet haben, sollten trotzdem den nächsten Abschnitt nicht überspringen, da der IEC-Bus mehrere schwierige technische Einzelheiten gleichzeitig enthält, die man alle verstehen sollte, wenn man sich mit dem Bau von Geräten mit einer IEC-Bus-Schnittstelle beschäftigen will.*

Elemente und Arbeitsweise eines Bus-Systems

Warum nur 1 Talker?

An den IEC-Bus lassen sich gleichzeitig verschiedene Geräte anschließen; diese können *Talker* – d. h. Sender – sein, sie können *Listener* – d. h. Zuhörer – sein, und sie können *Controller* (Steuergeräte) sein. Dabei ist es nicht erforderlich, daß ein Gerät nur Talker oder nur Listener ist. Es gibt jede nur denkbare Kombination aus diesen drei Eigenschaften. Aber eines steht fest: nur *ein* Controller zur Zeit darf das System steuern! Und nur *ein* Talker zur Zeit darf Daten auf den BUS senden! Erst wenn dieser Talker mit seiner Sendung fertig ist, darf ein zweiter Talker stattdessen das Senden übernehmen. Das einzige, was gleichzeitig von mehreren Stellen her erlaubt ist, ist das *Zuhören (Bild 6)*. Wenn nämlich außer dem Talker „T 1“ noch ein weiterer Talker gleichzeitig senden würde, so würden sich die Daten auf dem Bus gegenseitig stören. Wir sehen hieraus gleich eine Eigenschaft des Bus-Systems: alle Übertragungen müssen *nacheinander* (also *seriell*) erfolgen. *Gleichzeitig* lassen sich auf einem Bus nur die einzelnen Bits eines Zeichens übertragen, die jedes für sich eine eigene Leitung haben, d. h., für ein Bus-System gibt es mehrere Datenleitungen, die parallel nebeneinander liegen und die ein bitparalleles, zeichen-serielles Übertragen der Daten ermöglichen. Im Gegensatz zu der Beschränkung auf einen einzigen Talker gilt diese Beschränkung für das Empfangen von Daten, wie gesagt, nicht; es können gleichzeitig beliebig viele Empfänger eingeschaltet und empfangsbereit sein.

Aus *Bild 7* kann man sehen, warum nur ein Sender zur Zeit auf dem Datenbus senden darf: der Sender 1 hat als letzten

Baustein vor dem Datenbus z. B. ein NAND-Glied mit offenem Kollektor. Solange das Freigabesignal 1 auf Null liegt, hat der Baustein S 1 am Ausgang seinen High-Zustand [3]. Das bedeutet, daß der Sender S 1 den Bus nicht beeinflußt, da er hochohmig geschaltet ist. Auf dem Bus können jetzt andere Daten transportiert werden. Der Sender n hat als letzten Baustein z. B. einen Drei-Zustands-Logik-Baustein *(Tri-state-logic)*; dieser hat an seinem Ausgang einen hochohmigen Zustand solange die Freigabeleitung n auf Null geschaltet ist. Das bedeutet, daß auch dieser Baustein die Bus-Leitung nicht beeinflußt. Erst wenn die Freigabeleitung n auf „1" geschaltet ist, werden die Sendedaten aus dem Sender N auf den Bus gegeben. Das Ausgeben von Daten auf den Datenbus erfolgt jetzt, je nachdem, ob es über den offenen Kollektor oder einen Drei-Zustands-Baustein geht, auf zwei verschiedene Arten: Ein NAND-Glied mit offenem Kollektor zum Beispiel kennt im Grunde genommen nur einen Low-Zustand am Ausgang. Ein solches Glied ist entweder auf Low geschaltet und zieht den gesamten Datenbus auf Low, oder aber es ist nicht auf Low geschaltet, und dann muß über einen externen Arbeitswiderstand dafür gesorgt werden, daß der Datenbus auf High liegt. Wenn jetzt zwei Sender gleichzeitig auf demselben Bus arbeiten, so genügt es, daß ein einziger dieser Sender auf Low geschaltet ist, um den gesamten Datenbus auf Low zu ziehen. Wenn jetzt der zweite arbeitende Sender eigentlich ein High-Signal übertragen will, so wird dieses durch das Low-Signal des ersten Senders gestört. *Man sieht hieraus: um Daten über einen Datenbus ungestört übertragen zu können, darf immer nur ein Sender zur Zeit arbeiten!*

Schwieriger wird es bei Drei-Zustands-Logik-Bausteinen: Wenn wir davon ausgehen, daß ein Baustein hierbei ein Low-Signal, der zweite gleichzeitig ein High-Signal ausgibt, so werden diese Bausteine sogar noch *überlastet*, da ein Baustein die Bus-Leitung niederohmig auf Low zieht, während der andere die Bus-Leitung auf High zu ziehen versucht. Dadurch fließen starke Querströme von einem Baustein über den anderen.

Hingegen spielt es keine Rolle, wieviele Empfänger gleichzeitig am Bus anliegen. Hier gibt es nur eine Belastungsgrenze *(fan-out)*, denn wenn man zu viele Empfänger gleichzeitig an den Bus anschließt, so wird der Sender nicht mehr in der Lage sein, den Strom für die Empfänger gleichzeitig zu liefern.

„Active-low" oder „Active-high"?

Anhand des Bildes 7 soll noch eine weitere Eigenschaft des Busses erklärt werden: wenn die Sendedaten 1 über den Baustein S 1 auf den Datenbus ausgegeben werden – d. h. die Freigabeleitung 1 ist auf High geschaltet – so werden die Daten auf dem Datenbus invertiert übertragen. Daten, die innerhalb des Gerätes High waren, sind auf dem Bus jetzt Low und umgekehrt. Wenn wir dagegen die Sendedaten n aus dem Sender mit Drei-Zustands-Logik-Baustein ansehen, so werden sie in diesem nicht invertiert, d. h. ein High-Signal aus dem Sender n wird im High-Zustand über den Datenbus übertragen, und ein Low-Signal wird als Low-Signal übertragen.

Es muß also für eine Schnittstelle definiert sein, ob ein „wahres" Signal (1-Signal) als *„active high"*- oder als *„active low"*-Pegel übertragen wird.

Welche Begrenzungen gibt es noch?

Nachdem wir aus Bild 7 schon gesehen haben, daß die Zahl der Empfänger wegen der Leistungsfähigkeit der Sender begrenzt ist (Begrenzung durch Stromgrenze der Bausteine), soll jetzt hier noch kurz über die erreichbare Länge eines solchen Bus-System gesprochen werden; sie wird begrenzt durch die

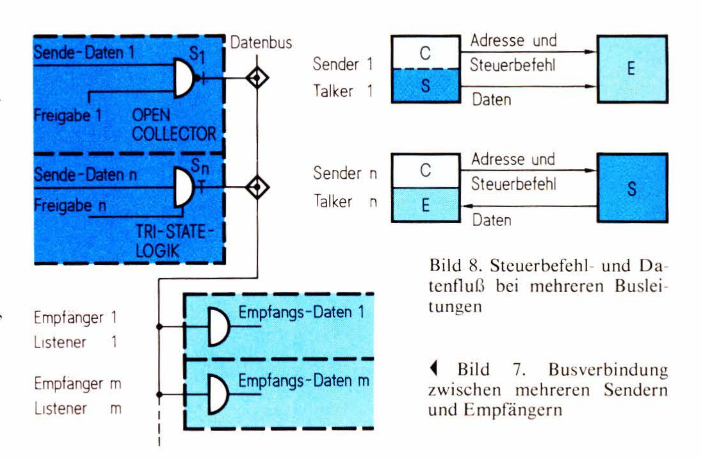


Bild 8. Steuerbefehl- und Datenfluß bei mehreren Busleitungen

◀ Bild 7. Busverbindung zwischen mehreren Sendern und Empfängern

Kapazität und den Leitungswiderstand des Kabels. Je länger eine Verbindungsleitung wird, desto größer wird ihre Kapazität. Ein Verbindungskabel mit großer Kapazität bedeutet aber, daß die Impulse, die über diesen Bus übertragen werden können, länger sein müssen; sind sie aber zu lang, dann kann man nicht mehr genügend Daten schnell genug von einem Gerät zum anderen übertragen.

Der Verlustwiderstand der Drähte liegt praktisch zwischen den Senderausgängen und den Empfängereingängen. Bei zu großer Leitungslänge stellt dieser Widerstand einen zusätzlichen Spannungsabfall für die Signale dar; durch ihn wird die Störsicherheit des Systems ungünstig beeinflußt, da der Low-Pegel der Empfänger immer weiter weg von Null Volt liegt, je mehr Widerstand sich zwischen den Empfängereingängen und den Senderausgängen befindet.

Wie seriell ist ein Bus-System?

Bei dem Bus-System von Bild 4 waren noch eine ganze Reihe von verschiendenen Bussen parallel nebeneinander vorhanden (siehe auch *Bild 8*). aus dem Steuergerät C (*Controller*) kommen der Adreßbus und der Steuerbus sowie die Zeit-Signale. Über das Bus-System gelangen diese Signale an den Empfänger E und dieser übernimmt Daten, die aus dem Sender S kommen. Hierbei gibt es zwei Möglichkeiten: das Steuergerät selbst ist Sender und sendet die Daten an einen externen Empfänger, oder aber das Steuergerät steuert nur die Ausgabe der Daten aus einem externen Sender und übernimmt diese Daten als Empfänger. In einem solchen Bus-System laufen Adreßbus und Steuerbus über zusammen 8 Leitungen, während der Datenbus weitere 8 Leitungen benötigt. Hierzu kommen noch einige zusätzliche Leitungen, über die die Zeit-Signale übertragen werden.

Wenn man die Zahl der Leitungen verringern will, wie es z. B. auch beim IEC-Bus der Fall ist, so gibt es hierfür folgende Möglichkeiten: da wir auf Adreß- bzw. Steuersignale in einem Bus-System nicht verzichten können, kann man die Übertragung von Adressen und Daten nur zeitmultiplex *nacheinander* über die gleiche Leitung vornehmen *(Bild 9)*. In diesem Fall muß über eine zusätzliche Leitung angekündigt werden, ob die über den Datenbus übertragenen Signale Adressen oder Daten sind. Innerhalb des Steuergerätes muß dafür ein „Adreß-Signal" erzeugt werden. Das Steuergerät legt die Adreß-Signal-Leitung auf High, wenn die *Adresse* übertragen wird.

Im Empfänger wird, wenn die Adreß-Signal-Leitung auf High liegt, der Übertragungskanal auf ein Eingabetor geschaltet, an dessen Ausgang wieder die Adresse zur Verfügung steht.

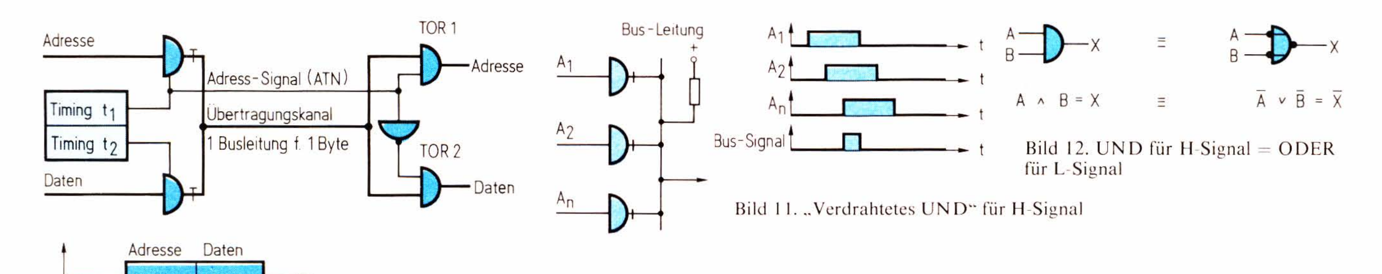


Bild 9. Bus-System für 1 Byte parallel

Bild 11. „Verdrahtetes UND" für H-Signal

Bild 12. UND für H-Signal = ODER für L-Signal

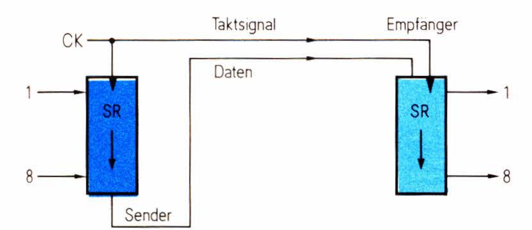


Bild 10. Serielles Bus-System für 8 Bit

Danach werden die Daten über den gleichen Übertragungskanal übertragen. Da das Adreß-Signal während der Datenübertragung auf Low liegt, wird jetzt über einen Inverter das zweite Tor im Empfänger aufgeschaltet; an seinem Ausgang stehen dann die Daten zur Verfügung. Da bei einer solchen Übertragungsweise Adresse und Daten niemals gleichzeitig zur Verfügung stehen, müssen in den angeschlossenen Geräten die übertragenen Adressen gespeichert werden. *Nur wenn die Adresse in einem Speicher zwischengespeichert ist, weiß das adressierte Gerät, daß es die übertragenen Daten annehmen soll.*

Beim IEC-Bus gibt es ein entsprechendes Adreß-Signal, es heißt „*Attention*"-ATN. Durch die Verwendung dieses Signals ist es möglich, die Zahl der Adreß- und Daten-Leitungen, die vorher 16 war, auf insgesamt *9 Leitungen* zu reduzieren. Nicht verringert wird hierdurch die Zahl der sonst noch erforderlichen Leitungen, die zum Übertragen der Zeit-Signale gehören. Was man hierunter im einzelnen zu verstehen hat, wird später erklärt. *Ein solches Übertragungssystem nennt man ein Bit-paralleles/Byte-serielles Übertragungssystem.*

Im Gegensatz dazu versteht man unter einem Bit-seriellen/Byte-seriellen System das, was beim Fernschreiber für die Verbindung von zwei Geräten angewendet wird. In *Bild 10* ist hierfür ein Schema gegeben. Die 8 Leitungen, die für ein Bit-paralleles System erforderlich sind, werden dabei in ein Schieberegister SR gegeben und mit Hilfe von Taktsignalen (*clock* CK) seriell aus diesem über eine Leitung ausgeschoben. Die gleichen Taktsignale dienen zum Einschieben der seriellen Signale in einen Empfänger. Nach 8 Takten sind die 8 parallelen Bits des Senders wieder in den Empfänger eingeschoben und stehen parallel zur Verfügung. In Bild 10 ist eine zweite Übertragungsleitung für die Taktsignale eingezeichnet.

Ein solches Übertragungssystem nennt man ein *synchrones Übertragungssystem*, da die Daten gleichzeitig mit den dazugehörigen Taktsignalen übertragen werden. Beim Fernschreiber, der die Datenübertragung prinzipiell auf diese Art ermöglicht, handelt es sich allerdings um ein *asynchrones System;* hier gibt es keine Taktsignale. Die zu einem Zeichen gehörenden Bits werden dabei in einem bestimmten Zeitraster übertragen und durch Start- und Stopschritte gekennzeichnet. Der Empfänger ist in der Lage, sich durch Überwachung von Start- und Stopschritten auf ein laufendes Signal einzupendeln.

Nun ist die Beschreibung eines vollseriellen Bus-Systems (Bit-serielles-Byte-serielles System), das über eine 2-Draht-Leitung wie beim Fernschreiber funktioniert, damit noch nicht abgeschlossen. Bei dem Bit-parallelen Bus-System von Bild 9 war ja bereits eine zusätzliche Adreß-Leitung für das Signal ATN erforderlich. Ferner gab es noch eine ganze Reihe Leitungen, die der Übertragung von zeitlichen Zuständen diente *(Timing)*, und außerdem befindet sich normalerweise in einem solchen Bus-System, z. B. auch im IEC-Bus-System, eine Reihe von Hilfsleitungen, die bisher noch gar nicht erwähnt wurden und auf die später erst eingegangen wird. Allein die Übertragung des Adreß-Signals ist in einem Fernschreibkanal nicht so ohne weiteres möglich.

Trotzdem gibt es durchaus Möglichkeiten, ein voll-serielles Bus-System zu bauen. Da hier aber eine Einführung in den IEC-Bus gegeben werden soll, wird von einer näheren Erläuterung der dabei auftretenden Schwierigkeiten und wie man sie lösen könnte, abgesehen. Zur Zeit laufen verschiedene Untersuchungen, ob sich eine ähnliche Norm – wie es der jetzt bekannt gewordene IEC-Bus ist – auch für ein Bit- und Byte-serielles Bus-System finden läßt.

„Verdrahtetes UND" und „Verdrahtetes ODER"

Jeder, der sich mit digitaler Elektronik beschäftigt, kennt die dabei benutzten Grundbausteine: es gibt UND-Glieder, ODER-Glieder und Umkehrstufen (Inverter). Aus diesen Grundelementen lassen sich dann höhere Bauelemente zusammenbauen, z. B. Flipflops, aus diesen dann wieder Zähler, Schieberegister und Speicher. Außer diesen aus elektronischen Bauelementen herstellbaren Logikbausteinen gibt es noch das sogenannte „verdrahtete UND" *(Wired-AND)*; dies ist kein Logik-Baustein, sondern nur eine logische Verknüpfung, die durch die Verdrahtung der Bauelemente hergestellt wird *(Bild 11)*. Verbindet man die Ausgänge von Bausteinen mit offenem Kollektor miteinander und legt diese Bus-Leitung über einen gemeinsamen Arbeitswiderstand auf Plus, so ist diese Bus-Leitung dann und nur dann auf High-Signal, wenn alle Ausgänge der angeschlossenen Bausteine auf High-Signal liegen. Ist nur ein einziger Ausgang auf Low-Signal, so ist die gesamte Bus-Leitung auf Low. Das eben Gesagte gilt für normale TTL-Bausteine mit positiver Logik (*active high*).

Das bedeutet aber, daß die Bus-Leitung in der in *Bild 11* dargestellten Form eine UND-Verknüpfung für die Signale $A_1 ... A_n$ darstellt (siehe auch Pegel-Diagramm).

In *Bild 12* ist ein UND-Glied mit seiner Verknüpfungsformel dargestellt. Der daneben dargestellte ODER-Baustein mit den Invertierungspunkten an allen Ein- und Ausgängen ist in seiner Funktion dem UND-Baustein gleich. Denn es gilt: $A \wedge B = X$; es gilt aber auch: $\overline{A} \vee \overline{B} = \overline{X}$, und das bedeutet, daß ein Null-Signal am Ausgang wieder erscheint, wenn der eine *oder* andere Eingang ein Null-Signal sieht. *Jeder UND-Baustein für positive Signale ist gleichzeitig ein ODER-Baustein für Null-Signale.*

In *Bild 13* ist ein „verdrahtetes ODER" für Low-Signale dargestellt. Über die Inverter-Ausgangsstufen werden die Signale $A_1 ... A_n$ auf den Bus geschaltet; wenn eines dieser Signale „wahr" ist, so liegt am Ausgang des dazugehörigen Sende-

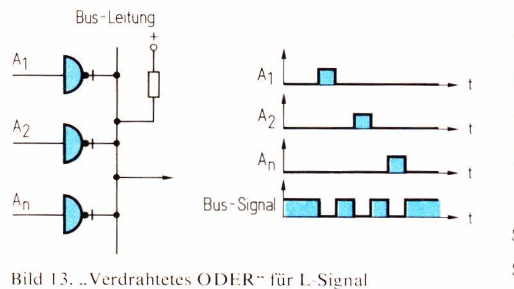


Bild 13. „Verdrahtetes ODER" für L-Signal

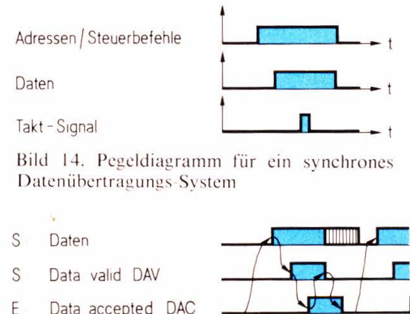


Bild 14. Pegeldiagramm für ein synchrones Datenübertragungs-System

Bild 16. Asynchrones Zweidraht-Handshake: Aufeinanderfolge der einzelnen Flanken

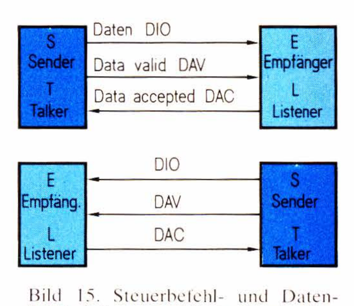


Bild 15. Steuerbefehl- und Datenfluß bei einem Zweidraht-Handshake-System

bausteins ein Low-Signal vor, und damit befindet sich die gesamte Bus-Leitung im Low-Zustand. Es spielt keine Rolle mehr, ob gleichzeitig noch weitere Sendesignale zur Verfügung stehen oder nicht. Ein einziges Signal genügt, um den gesamten Bus auf Low zu ziehen. Das aber ist die Funktion, die man üblicherweise ODER nennt. *Wir merken uns, daß auf einer Bus-Leitung, d. h. auf einer einfachen Drahtverbindung, ein verdrahtetes UND für High-Signale und ein verdrahtetes ODER für Low-Signale möglich ist!*

Das Zweidraht-Handshake-Verfahren

Früher verwendete man vorwiegend synchrone Datenübertragungssysteme, das bedeutet, daß zusätzlich zu den Daten noch ein Taktsignal übertragen wurde; mit diesem wurde dem Empfänger die Gültigkeit der Daten mitgeteilt. Ein solches synchrones Übertragungssystem ist der Zusammenschluß von zwei Geräten nach Bild 1, der Anschluß von Meßgeräten an einen Koppler nach Bild 2 und auch das in Bild 4 dargestellte Bus-System ist ein synchrones System mit einem Taktsignal *(Bild 14)*. Um an den Koppler von Bild 2 als Empfänger einen Fernschreiber anschließen zu können, müßte sich in dem Koppler bereits ein asynchrones Datenübertragungssystem befinden.

Außer synchroner und asynchroner Datenübertragung gibt es noch die Übertragung von Daten im *Handshake-Verfahren;* dieses ist ein asynchrones Datenübertragungssystem, bei dem die Daten mit einer variablen Geschwindigkeit übertragen werden können. Beim Fernschreiber war die Bit-Rate definiert, und der Empfänger konnte sich mit Hilfe von Start- und Stopschritten immer wieder neu auf die Frequenz des Senders synchronisieren. Beim *Handshake-Verfahren* ist für die Übertragung jedes Zeichens ein Melde- und ein Quittungssignal erforderlich *(Bild 15)*.

Ein einfaches *Handshake-Verfahren* ist das sogenannte Zweidraht-Handshake-Verfahren. Allerdings kann man mit ihm nur zwei Geräte aneinander anschließen. Hierbei wird das eine Gerät als Sender, das andere als Empfänger benutzt. Bei der IEC-Bus-Norm z. B. würde der Sender *Talker* heißen und der Empfänger ***Listener***. Die Datenübertragung geschieht hier Bit-parallel/Zeichen-seriell über die Datenleitung DIO *(Data IN/OUT)*. Jedes Zeichen, das übertragen wird, wird durch eine Meldung auf der ***Data-Valid-Leitung*** DAV angezeigt; der Empfänger überwacht sie und übernimmt das Zeichen, wenn die Daten zur Verfügung stehen und er bereit ist, sie zu übernehmen. Wenn der Empfänger die Daten übernommen hat, quittiert er sie über die ***Data-Accepted-Leitung*** DAC. Durch dieses Quittungssignal weiß der Sender, wann er aufhören kann, die alten Daten zu senden, bzw. wann er neue Daten senden darf. Daher übrigens die Bezeichnung *Handshake:* ein Mensch reicht dem anderen die Hand, dieser reicht seinerseits seine Hand zurück. Eine Möglichkeit der Datenanforderung wird später beschrieben.

Das *Handshake-Verfahren* gilt im beschriebenen Fall nur für den Zusammenschluß von zwei Geräten, allerdings kann die Sende- bzw. Empfängereigenschaft hierbei umgeschaltet werden. ***Das bedeutet, der Sender kann zum Empfänger und der Empfänger zum Sender werden. Hierbei kehren sich alle Signalrichtungen um*** (Bild 15).

Ein Pegeldiagramm zu Bild 15 ist in *Bild 16* dargestellt. Ein Handshake-Zyklus beginnt damit, daß der Sender Daten auf den Datenbus ausgibt; anschließend meldet er diese Daten über die Data-Valid-Leitung als gültig. Der Empfänger überwacht die Data-Valid-Leitung, und wenn er das Signal ***„Data-Valid"*** erkennt, übernimmt er die Daten und gibt über die Data-Accepted-Leitung die Rückmeldung DAC, daß er die Daten empfangen hat. Der Sender seinerseits überwacht die Data-Accepted-Leitung, und wenn die Rückmeldung kommt, daß die Daten empfangen worden sind, so nimmt der Sender das Data-Valid-Signal DAV weg und kann kurz danach auch die Daten wegnehmen, da der Empfänger diese Daten nicht mehr benötigt. Der Empfänger, der ja die Data-Valid-Leitung überwacht, kann jetzt von sich aus zu einem beliebigen Zeitpunkt danach das Data-Accepted-Signal wegnehmen. Danach beginnt ein neuer Zyklus, in dem das nächste Zeichen übertragen werden kann.

Nach dieser prinzipiellen Darstellung eines Handshake-Zyklus folgt in *Bild 17* eine noch genauere Beschreibung der einzelnen Signale. Die Daten werden als erstes ausgegeben und auf der Data-Valid-Leitung gültig gemeldet. Wenn der Empfänger auf die Gültigmeldung DAV jetzt mit einem Data-Accepted-Signal DAC reagiert, so weiß der Sender durch die Flanke des DAC-Signals, daß die Daten übernommen worden sind. Die Zeit, in der die Data-Accepted-Leitung in dem Zustand „Übernommen" verbleibt, ist die interne Verarbeitungszeit des Empfängers; denn es ist möglich, daß der Empfänger die Daten mit der Flanke übernimmt, aber nicht sofort weitere Daten gebrauchen kann.

Nach der Übernahmeflanke des DAC-Signals kann der Sender das DAV-Signal ausschalten und außerdem die Datenausgabe beenden, d. h. die Datenleitung ist *passiv*, und die

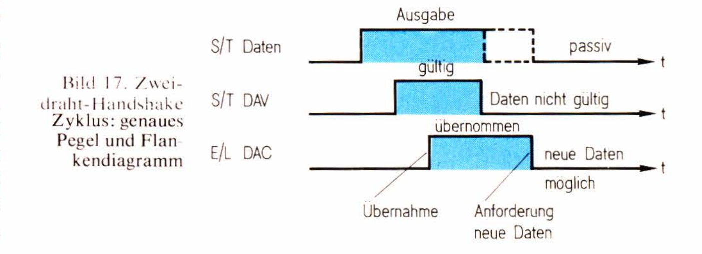


Bild 17. Zweidraht-Handshake Zyklus: genaues Pegel und Flankendiagramm

Data-Valid-Leitung DAV liegt in ihrem Zustand, der meldet, daß die Daten nicht gültig sind. Der Empfänger schaltet das Data-Accepted-Signal erst dann aus, wenn er neue Daten gebrauchen kann. Die Rückflanke des Data-Accepted-Signals ist hier also eine Anforderung für neue Daten!

Dieser Zweidraht-Handshake funktioniert für zwei Geräte in beliebiger Richtung mit Umschalten der Übertragungsrichtung. Das Umschalten selber darf hierbei natürlich nur zwischen zwei Übertragungszyklen passieren, d. h. in einem Moment, wo Daten nicht gültig sind und neue Daten möglich sind (passiver Bus-Zustand). Ein Umschalten innerhalb eines Zyklus ist nicht möglich.

Wenn man, wie beim IEC-Bus realisiert, mehr als zwei Geräte gleichzeitig miteinander verbinden will, so ist der Zweidraht-Handshake der eben beschriebenen Bauart nicht mehr ausreichend. Die einfachste Konfiguration mit mehr als zwei Geräten ist gegeben, wenn ein Talker mit zwei Listenern arbeiten soll (Bild 18); alles, was hier für das kleinste System von einem Talker und zwei Listenern gezeichnet ist, gilt auch für weitere Talker, die am gleichen System angeschlossen, aber passiv sind, und es gilt genauso für weitere Listener-Geräte.

Als erstes wollen wir uns klarmachen, warum der Zweidraht-Handshake für ein System mit mehr als zwei Geräten nicht mehr brauchbar ist: In dem Pegeldiagramm von Bild 18 sind die beiden Data-Accepted-Ausgänge der Listener L 1 und L 2 getrennt dargestellt. Dabei wird angenommen, daß die beiden Listener verschiedene Zeiten für die Verarbeitung der empfangenen Daten benötigen. Wie schon gesagt, setzt der Sender als erstes die Daten und meldet sie über die Data-Valid-Leitung als gültig. Wenn jetzt der erste Listener sehr schnell auf diese Gültigmeldung reagiert und sein Data-Accepted-Signal gibt, so nützt das allein noch gar nichts. Erst nach der Verzögerungszeit „t 1", wenn auch der zweite Listener seinen Data-Accepted-Ausgang in High-Zustand bringt, wird durch die UND-Funktion auf dem Bus (verdrahtetes UND) der Data-Accepted-Bus in den High-Zustand gebracht. Der Talker hat auf diesen High-Zustand auf dem Data-Accepted-Bus gewartet und nimmt jetzt das Date-Valid-Signal und die Daten weg. Daraufhin nehmen die Listener L 1 und L 2 im ersten Zyklus gleichzeitig ihre Data-Accepted-Signale weg, und der DAC-Bus geht auf Null und signalisiert dem Talker damit, daß der nächste Zyklus beginnen kann.

Das funktioniert noch; aber im zweiten Zyklus gibt es Schwierigkeiten: die Daten werden auf den Datenbus gegeben, und die Data-Valid-Leitung geht in den High-Zustand, womit die Daten gültig gemeldet werden. Jetzt ist von der Voraussetzung ausgegangen worden, daß beide Listener die Daten sofort übernehmen, d. h. DAC 1 und DAC 2 gehen sofort in den High-Zustand; danach kann der Talker die DAV-Leitung auf Low setzen. Hat er das getan, reagiert der Listener 2 sehr schnell und setzt seinen Ausgang DAC_2 auf Null; damit wird der DAC-Bus auf Null gesetzt, *obwohl der Listener 1 an seinem Ausgang noch ein High-Signal abgibt*, weil der Listener 1 noch nicht in der Lage ist, neue Daten zu übernehmen. Erst nach einer Zeit „t 2" ist der Listener 1 auch in der Lage, neue Daten aufzunehmen. Nachdem aber der DAC-Bus schon vorher auf Low gegangen ist, glaubt der Talker, er könnte neue Daten auf den Bus geben; denn ein Low-Signal auf dem DAC-Bus bedeutet ja eine Anforderung neuer Daten. Der Talker setzt daraufhin neue Daten auf den Datenbus, und wenn er jetzt schnell genug ein Data-Valid-Signal gibt, bevor der Listener 1 überhaupt ein Data-Accepted-Signal gegeben hat, so werden die Daten aus dem dritten Zyklus nicht mehr in den Listener L 1 übernommen.

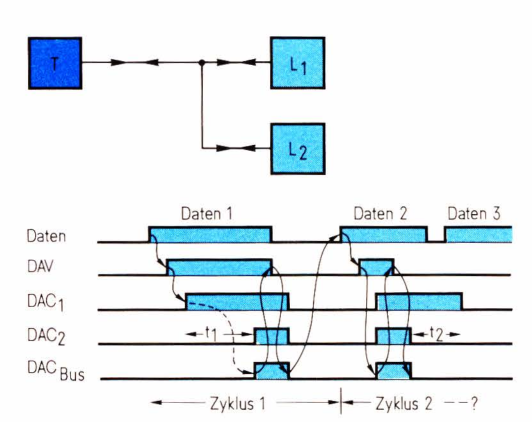


Bild 18. Hand-shake bei mehreren Empfängern (Listener L 1 und L 2)

Im beschriebenen Fall war beim Pegeldiagramm angenommen worden, daß Data-Accepted und Data-Valid jeweils auf High gesetzt werden, wenn die Signale „wahr" sind. Wenn man jetzt den Bus in Gedanken umpolt, d. h. also, die „wahren" Signale zu Low-Signalen macht, so ändert das an der Tatsache nichts, daß man keine zwei oder mehr Listener auf dem Bus gleichzeitig überwachen kann. Das liegt daran, daß die Data-Accepted-Leitung in unserem Fall zwei Funktionen hat: *erstens* meldet sie mit dem High-Pegel, daß die Daten übernommen sind und daß der Empfänger intern noch mit der Verarbeitung beschäftigt ist und im Moment noch keine neuen Daten übernehmen kann; *zweitens* signalisiert ein Low-Signal auf der Data-Accepted-Leitung die Anforderung neuer Daten.

Wenn jetzt nicht nur einer, sondern mehrere Listener auf diesem Bus angeschlossen sind, so muß man zwei UND-Verknüpfungen auf dem Bus haben, sowohl für das „Übernommen"-Signal DAC, als auch für das Signal, daß neue Daten möglich sind.

Die eine UND-Verknüpfung war bisher durch einen High-Pegel, die andere durch einen Low-Pegel auf der DAC-Leitung dargestellt. Aus den Erläuterungen zu Bild 11...13 wissen wir aber schon, daß man UND-Verknüpfungen auf einen Bus nur für High-Pegel vornehmen kann. Wenn man auf dem Bus eine UND-Verknüpfung dafür vorsehen will, daß alle angeschlossenen Listener die Daten übernommen haben, so muß man eine UND-Verknüpfung für das „Übernommen"-Signal einführen. Wenn man jetzt sichergehen will, daß neue Daten erst dann ausgesendet werden, wenn alle angeschlossenen Listener in der Lage sind, sie auch zu übernehmen, so muß man eine *getrennte* UND-Verknüpfung für die Datenanforderung auf dem Bus einführen. Deshalb ist im IEC-Bus außer den bisher beschriebenen beiden Leitungen Data-Valid und Data-Accepted noch eine *dritte* Bus-Leitung vorgesehen: Sie heißt *Ready-for-Data* und gibt „Grünes Licht" für neue Daten.

Wir erkaufen uns also die Möglichkeit, mehr als einen Listener mit dem jeweiligen Talker gleichzeitig verbinden zu können, durch Einführung einer dritten Handshake-Leitung.

Der Autor dankt allen Stellen der PHILIPS Elektronik Industrie GmbH, die ihn bei der Abfassung des Manuskripts unterstützt haben. Im zweiten Teil dieser Arbeit wird das Dreidraht-Handshake-Verfahren näher beschrieben, dann werden alle Leitungen und Signale des IEC-Bus erklärt, schließlich wird gezeigt, wie die Datenübertragung auf dem IEC-Bus durch die Übertragung von bestimmten Code-Zeichen als Steuerbefehl vor sich geht.

Literatur

[1] *Freytag, H. H.:* Ein Sieg der Vernunft (Leitartikel). ELEKTRONIK 1975, H. 1, S. 3.

[2] *Schmidt, U.:* Stand der Arbeiten am IEC-Bus-System. ELEKTRONIK 1975, H. 1, S. 58.

[3] *Ricci, D.W.* und *Nelson, G.E.:* Standard instrument interface simplifies system design. Electronics, Bd. 47 (1974), H. 23, S. 95...106.

Dipl.-Ing. Jürgen Klaus

Wie funktioniert der IEC-Bus?

2. Teil

Der erste Teil dieser Arbeit führte Schritt für Schritt in die Technik der Verkettung von Meßgeräten und von Ein-/Ausgabegeräten über digitale Sammelschienen ein, kurz Bus genannt und nunmehr in einem IEC-Normendokument international festgelegt. Die Besprechung ging von der elementaren Zusammenschaltung zweier Geräte („Sender" und „Empfänger") aus und reichte bis zum Zweidraht-Handshake-Verfahren. Im vorliegenden und abschließenden zweiten Teil wird das Dreidraht-Handshake-Verfahren erläutert, um dann auf Leitungen, Signale, Übertragungsmethoden und Codierung beim IEC-Bus einzugehen.

Das Dreidraht-Handshake-Verfahren

Der IEC-Bus benutzt ein Dreidraht-Handshake-Verfahren zur Datenübertragung. In *Bild 19* sind ein Zweidraht-Handshake und ein Dreidraht-Handshake so untereinander gezeichnet, daß die entsprechenden Pegel-Änderungen an der gleichen Stelle eintreten. Wir wollen kurz zusammenfassen, wie der Dreidraht-Handshake funktioniert:

Der Talker oder Sender wartet darauf, daß der Listener (oder Empfänger) auf der Ready-for-Data-Leitung RFD ein Ready-Signal abgibt und daß die Data-Accepted-Leitung DAC in der Low-Stellung liegt, d. h. der Empfänger hat keine Daten empfangen, die er noch nicht verarbeitet hat. Wenn diese Vorbedingungen erfüllt sind, kann der Sender die Daten auf den Bus schicken. Nach einer kurzen Einschwingzeit meldet der Sender diese Daten auf der Data-Valid-Leitung DAV gültig. Sowie der Empfänger das DAV-Signal empfängt, nimmt er das Ready-for-Data-Signal weg und meldet damit dem Sender, daß er vorläufig noch keine neuen Daten gebrauchen kann. Nach einer kurzen Verzögerungszeit, die vom inneren Aufbau des Empfängers abhängt, hat er diese Daten übernommen und meldet über die Data-Accepted-Leitung diese Tatsache an den Sender zurück. Daraufhin kann der Sender die Daten und das Data-Valid-Signal vom Bus wegnehmen. Wenn das Data-Valid-Signal verschwunden ist, nimmt der Empfänger auch das Data-Accepted-Signal weg; damit ist ein Übertragungszyklus beendet und der Bus ist in seinem Ruhezustand.

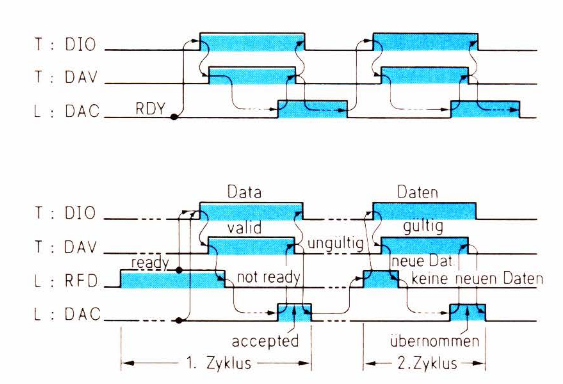


Bild 19. Gegenüberstellung von Zweidraht-Handshake (oben) und Dreidraht-Handshake (unten)

Wenn der Empfänger jetzt das nächste Zeichen aufnehmen kann, meldet er dies über die Ready-for-Data-Leitung und der nächste Zyklus kann beginnen. In Bild 19 ist dies dargestellt, wobei die Übergänge von einem Signal zum anderen durch Pfeile eingezeichnet sind.

Die Pegel

Die Datenübertragung im IEC-Bus geschieht mit TTL-Pegeln, die Ausgangsbausteine sollen normale Open-Collector-TTL-Bausteine sein, und die Eingangsbausteine (Empfänger) ebenfalls normale TTL-Bausteine. Um auf dem Bus die Reflexionen auf der Leitung niedrig zu halten, wird er in jedem Gerät entsprechend dem Wellenwiderstand des Kabels mit Widerständen abgeschlossen, d. h. an jeder Leitung wird in jedem Gerät ein Widerstand nach „Plus" und einer nach „Null Volt" gelegt. Da diese Widerstände eine zusätzliche Belastung für die Sender-Bausteine darstellen, werden endgültig als Bustreiber Open-Collector-Glieder mit einem Fan-Out von 30 eingesetzt. Die Norm sieht außerdem auch die Benutzung von Tri-State-Bausteinen vor. Hierauf soll aber in diesem Beitrag nicht näher eingegangen werden.

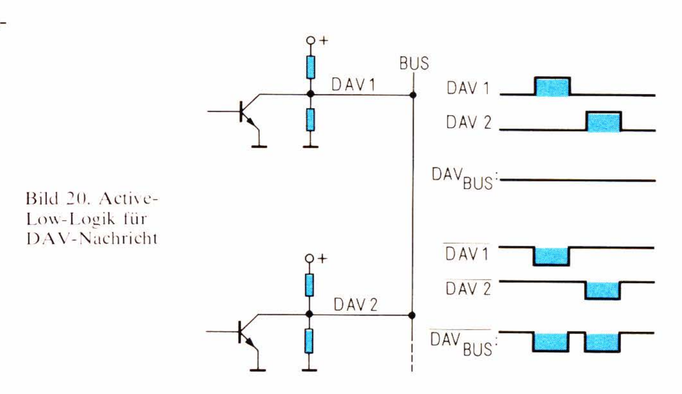


Bild 20. Active-Low-Logik für DAV-Nachricht

Als nächstes muß festgelegt werden, ob die Datenübertragung mit Active-Low- oder mit Active-High-Signalen erfolgen soll *(Bild 20)*. Als Beispiel diene die Übertragung des Signals DAV: Aus einem Sender soll das Signal DAV 1 auf den Bus gegeben werden können, aus einem zweiten Sender das Signal DAV 2. Wie wir früher schon festgestellt haben, bewirkt der Bus eine UND-Verknüpfung für High-Signale und gleichzeitig eine ODER-Verknüpfung für Null-Signale. Wenn das DAV-Signal nun wahlweise aus dem einen *oder* dem anderen Sender kommen soll, so brauchen wir auf dem Bus eine ODER-Verknüpfung; sie läßt sich nur erreichen, wenn man das DAV-Signal auf dem Bus als Active-Low-Signal betreibt. Wenn man sich jetzt alle anderen Signale, die auf dem Bus vorkommen können, ansieht, so gilt das, was wir gerade für DAV festgestellt haben, für fast alle anderen Signale ebenfalls: Jedes Signal muß von jedem Gerät gesendet werden können. *Das bedeutet aber, daß die Nachrichtenübertragung auf diesem Bus nur durch Low-Signale erfolgen kann!*

Genau gesagt, ist es bei Open-Collector-Bausteinen als Sender so, daß man einen Ruhezustand definieren muß; dieser kann nur ein High-Signal sein. Der aktive Zustand, mit dem eine bestimmte Aussage gemacht wird, kann dann nur ein in Richtung „Low" gehendes Signal sein, wenn diese Nachricht von allen Sendern wahlweise aussendbar sein soll.

Was hier gesagt wurde, daß nämlich die Übertragung von bestimmten Nachrichten mit Active-Low-Signalen erfolgen muß, gilt für alle Signale, die auf dem Bus durch eine ODER-Verknüpfung miteinander verbunden werden müssen. Die reine Datenübertragung, d. h. also die Pegel auf den 8 Datenleitungen, bleibt hiervon im Prinzip unberührt; man müßte nur dafür sorgen, daß alle Sender, die nicht aktiv tätig sind, in den Ruhezustand, d. h. in den High-Zustand gehen und nur der Sender, welcher auch wirklich Daten aussendet, den Low-Zustand annehmen kann. *Um auf dem Bus einheitliche Signale und Bezeichnungen zu schaffen, hat man sich in der Norm darauf geeinigt, daß die Daten auf dem Bus als Aktive-Low-Signale übertragen werden sollen.* Diese Festlegung gilt für alle Signale, die im IEC-Bus vorkommen.

Dies ist eine wichtige Tatsache, die man sich unbedingt klarmachen muß: Alle Signale auf dem IEC-Bus sind Active-Low-gehende Signale und ihre Bezeichnung gilt für den Active-Low-Zustand. Daher ist das Signal DAV auf dem Bus ein Active-Low-Signal und die Daten sind Valid, d. h. die Daten stehen zur Verfügung, wenn die DAV-Leitung Null-Pegel hat! Hat die Leitung High-Pegel, dann stehen keine Daten zur Verfügung.

Die Leitungen und Signale des IEC-Busses

Bisher haben wir die 8 Datenleitungen, die Data-Valid-Leitung, die Ready-for-Data-Leitung und die Data-Accepted-Leitung behandelt. Hierbei dienen die 8 Datenleitungen DIO zur Übertragung aller Daten, Adressen und Steuerbefehle. Die drei Leitungen DAV, RFD und DAC dienen als Dreidraht-Handshake-Leitungen. Im Text zu Bild 9 haben wir auch die Attention-Leitung ATN kennengelernt, mit der festgestellt wird, ob es sich bei der Nachrichtenübertragung um Daten oder Adressen handelt. Die „Adressen" – d. h. die zusammen mit dem Signal Attention ATN übertragenen Nachrichten – können wirkliche Geräteadressen sein, es kann sich dabei aber auch um spezielle Steuerbefehle handeln, auf die später noch eingegangen wird.

Außer den bisher schon beschriebenen gibt es im IEC-Bus noch folgende weitere Leitungen:

● *Interface clear IFC*

Über diese Leitung kann (nur!) das System-Steuergerät (System Controller) alle angeschlossenen Interfaces in ihre Grundstellung bringen; nach dem Einschalten der Netzspannung z. B. verschafft sich das Steuergerät mit diesem Signal sofort eine Ausgangslage, in der alle Interfaces in allen angeschlossenen Geräten in ihrer Grundstellung stehen und keines derselben versehentlich noch adressiert ist.

● *Remote enable REN*

Meßgeräte sind außer durch Fernbedienung über eine Bus-Leitung auch noch über Bedienelemente von der Frontplatte her einstellbar. Über das Signal REN werden alle Gerätefrontplatten zwangsweise in ihrer Bedienungsfunktion gesperrt, d. h. das Gerät ist gezwungen, auf Signale vom Bus her zu reagieren. Das Signal REN kann ebenfalls nur vom System-Steuergerät ausgesendet werden.

● *Service-Request SRQ*

Diese Leitung stellt eine Art Interrupt-Leitung dar. Jedes angeschlossene Gerät, welches in der Lage ist, Service-Request anzufordern, kann die SRQ-Leitung auf Low ziehen. („ODER" für Null-Signal). Das Steuergerät muß in der Lage sein, ein solches Service-Request-Signal zu erkennen und anschließend muß es feststellen, welches Gerät diese Anfrage ausgesendet hat; nach dieser Feststellung kann das Steuergerät dieses Gerät bedienen.

Als Beispiel eine Service-Request-Anforderung durch ein Interface für eine Fernschreibmaschine: wenn jemand von außen ein Zeichen auf den Fernschreiber gibt, so muß das Steuergerät wissen, daß es aus dem IEC-Bus-Interface ein eingegebenes Zeichen abholen muß. Das IEC-Bus-Interface könnte sich mit SRQ melden und nachdem das Steuergerät festgestellt hat, daß diese Service-Anforderung vom Fernschreib-Interface kommt, kann es sich das eingegebene Zeichen von dort abholen.

● *End-or-Identify-Signal EOI*

Das EOI-Signal hat, wie der Name sagt, zwei Funktionen: erstens wird mit ihm das *Ende* einer Blockübertragung angezeigt, und zweitens wird es bei der *Identifizierung* von Geräten, die eine Service-Request-Anfrage gestartet haben, benutzt. Das „Ende"-Signal wird vom Talker zusammen mit dem letzten Zeichen eines Datenblocks übertragen und dient für das Steuergerät zur Erkennung eines Blockendes; danach weiß das Steuergerät, daß es neue Talker und/oder Listener adressieren muß oder zumindest eine neue Datenblock-Übertragung einleiten muß.

Das *Identify*-Signal wird vom Steuergerät ausgesendet und dient dazu, ein Gerät, das Service-Request ausgesendet hat, zu einem Identify-Zyklus aufzufordern. Das EOI-Signal muß man in Verbindung mit dem ATN-Signal betrachten:

ATN	EOI	DIO
0	0	normales Datenbyte
0	1	letztes Datenbyte eines Blockes (END)
1	0	Adresse, Universal-Befehl, Adressierter Befehl
1	1	Aufforderung zur Identifizierung nach Service Request-Anfrage (IDENTIFY)

0 bedeutet: Bus nicht aktiv, High-Pegel!
1 bedeutet: Bus aktiv, Low-Pegel!

Wie bei allen guten Regeln gibt es leider auch beim IEC-Bus eine *Ausnahme* von dem Prinzip, daß alle Signale Active-Low-Signale sind: da die Data-Accepted-Leitung und die Ready-for-Data-Leitung als einzige Leitungen auf dem Bus eine UND-Verknüpfung bilden müssen, funktioniert die Idee mit den standardisierten Active-Low-Pegeln hier nicht mehr. *Die Bezeichnung „Data-Accepted" bzw. „Ready-for-Data" muß sich hier auf den High-Pegel beziehen!* Und um die Verwirrung bei der Bezeichnung der einzelnen Leitungen nicht zu groß werden zu lassen, hat man die Namen für die Leitung Data-Accepted und Ready-for-Data geändert in „Not-Data-Accepted" und „Not-Ready-for-Data". Die Abkürzungen lauten dann NDAC und NRFD. In *Bild 21* sind diese neuen Namen den bisher benutzten gegenübergestellt: die Datenleitung DIO, die Data-Valid-Leitung DAV und auch alle anderen Leitungen, die in diesem Bild nicht mit eingezeichnet sind, sind Leitungen, wo Signale als Active-Low-Signale übertragen werden. Wenn man Data-Accepted und Ready-for-Data ebenfalls als Active-Low-Signal übertragen wollte, so würde sich ein Bild ergeben, wie es auf der linken Hälfte von Bild 21 dargestellt ist. Da aber Data-Accepted und Ready-for-Data eine UND-Verknüpfung auf dem Bus herstellen müssen, sind hier Active-High-Pegel erforderlich.

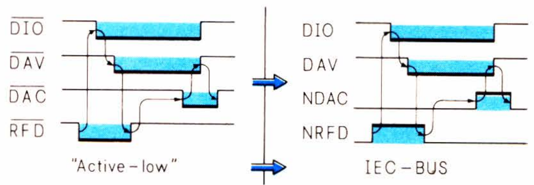


Bild 21. Dreidraht-Handshake mit „Active-Low"-Pegeln und, daraus abgeleitet, die Pegel beim IEC-Bus

In der rechten Hälfte von Bild 21 sind die endgültig festgelegten aktiven Pegel stärker dargestellt: bei DIO und DAV sind die Low-Pegel die eigentlich interessierenden Pegel, bei DAC und RFD sind in der rechten Hälfte die interessierenden Pegel dicker ausgezogen. Damit die Namensgebung für alle Leitungen gleich ist, nämlich für Low-Pegel, ist der Name von

DAC in Not-Data-Accepted NDAC, geändert, und der Name Ready-for-Data ist geändert in Not-Ready-for-Data NRFD.

Außerdem ist in der rechten Hälfte von Bild 21 noch eine Änderung vorgenommen worden, um die Anpassung an die Darstellung im IEC-Bus-Normen-Dokument zu erreichen; die Querstriche über den Namen sind weggelassen worden, auch wenn sich die Namen auf Active-Low-Pegel beziehen! Beim Lesen der IEC-Bus-Norm muß man sehr vorsichtig sein, und immer daran denken: *Alle Signale, die sich direkt auf den Bus* ***beziehen, sind Active-Low-Signale, auch wenn sie nicht mit Invertierungsstrichen versehen sind!*** Alle Signale innerhalb des Interfaces sind Active-High-Signale und sind deshalb natürlich auch nicht überstrichen.

Der Übergang vom IEC-Bus zur Interface-Elektronik

Wenn wir uns jetzt die wirklichen Bausteine, d. h. die Treiber und die Empfänger, in einem Interface ansehen, so ergibt sich die in *Bild 22* gezeigte Situation. Auf dem IEC-Bus sind alle Signale Low-Signale. Das bedeutet, die Datenleitungen DIO, die Data-Valid-Leitung DAV und die Sonderleitungen ATN, SRQ, EOI, REN und IFC gelangen über invertierende Empfängerbausteine in das Interface. Hinter diesen invertierenden Empfängerbausteinen liegen die Signale mit der *gleichen Benennung* vor, weil auf der Bus-Seite eine Active-Low-Logik vorliegt (auch wenn die Signale nicht überstrichen sind) und auf der Interface-Seite eine Active-High-Logik eingesetzt wird. Es ist sinnvoll, auf der Interface-Elektronik-Seite mit Active-High-Signalen zu arbeiten, da man den Aufbau zunächst aller Wahrscheinlichkeit nach mit TTL-Bausteinen ausführen wird.

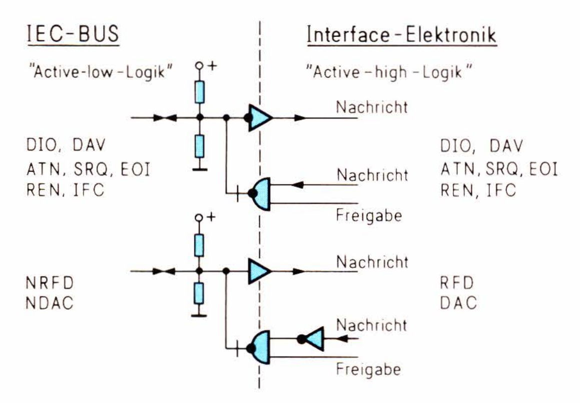


Bild 22. Bus-Treiber und Empfänger sowie Signalbezeichnungen auf dem Bus (Active-Low-Pegel) und in der Interface-Elektronik (Active-High-Pegel)

Wenn wir uns jetzt die gleichen Signale wie eben ansehen, aber diesmal in der umgekehrten Richtung, d. h. wie sie von dem Interface an den Bus ausgegeben werden, so benötigen wir hierfür einen invertierenden Baustein mit Open-Collector-Ausgang (Bild 22). Die Nachricht selber wird, wenn das Freigabesignal dies erlaubt, über ein Open-Collector-NAND-Glied auf den Bus ausgegeben. Nur wenn die Daten auch wirklich aus den Interface herauskommen sollen, wird über das Freigabesignal der Open-Collector-Baustein freigegeben. Sind die Daten nicht an der Übertragung beteiligt, so liegt das Freigabesignal auf Null und der Bus nimmt seinen High-Zustand an.

Ein wenig anders sieht es für die Signale Not-Ready-for-Data und Not-Data-Accepted aus. Die Signale NRFD und NDAC sind Low-gehende Signale auf dem Bus! (Das Signal Ready-for-Data bzw. Data-Accepted ohne „N" davor wäre ein High-gehendes Signal). Diese Signale gelangen von dem Bus über einen nicht invertierenden Baustein in das Interface. Das Signal heißt im Interface dann Ready-for-Data, d. h. diesmal ändert sich der Name, obwohl in dem Empfänger-Baustein selber keine Invertierung des Signals vorgenommen wird.

Tabelle 1. Leitungsbezeichnungen beim IEC-Bus

DIO	Data-Bus	Daten-Bus
DAV NRFD NDAC	Data-Byte Transfer-Control-Bus	Übergabe-Bus (Handshake-Bus)
ATN SRQ EOI REN IFC	General-Interface-Management-Bus	Steuer-Bus

Ebenso ist es bei der Ausgabe von Ready-for-Data und Data-Accepted aus einem Interface auf den Bus. *Die Nachricht selber wird nicht invertiert!* In Bild 22 ist das durch insgesamt zwei hintereinander liegende invertierende Bausteine dargestellt. Die Freigabe des Ausgangs-Sender-Bausteines erfolgt genau wie zuvor beschrieben über ein Freigabesignal. Wenn kein Ready-for-Data- oder Data-Accepted-Signal aus dem Interface heraus auf den Bus geliefert werden soll, dann liegt die Freigabeleitung auf Null und damit befindet sich der Ausgang des Sende-Bausteins im High-Zustand. Die Signale auf dem Bus werden dann nicht von diesem Interface beeinflußt, und auf dem Bus können Daten zwischen anderen Geräten ungestört hin und her transportiert werden. In *Tabelle 1* sind die Leistungsbezeichnungen des IEC-Busses in Kurzform, in englischer und in deutscher Sprache zusammengestellt.

Nachrichtenübertragung

Wenn ein solches System zu arbeiten anfängt, so wird als erstes das Steuergerät (System Controller) ein Interface-clear-Signal IFC an alle angeschlossenen Geräte versenden und damit die Interfaces in ihre Grundstellung bringen *(Bild 23)*. Als nächstes wird ein REN-Signal verschickt; es sorgt dafür, daß alle angeschlossenen Geräte von ihren Frontbedienelementen abgetrennt werden. Natürlich gibt es aber auch die Möglichkeit, diese Frontbedienung durch einen neuerlichen Fernsteuerbefehl wieder zu erlauben.

Nach diesen Vorarbeiten beginnt die eigentliche Übertragung von Adressen, Befehlen und Daten. In Bild 23 ist die Darstellung etwas anderes gewählt, als es in der IEC-Bus-Norm üblich ist, um das Verständnis zu erleichtern. Die Signalbezeichnungen enthalten in Klammern jeweils ein „H"- oder ein „L"-Zeichen, je nachdem ob das Signal durch ein High- oder durch ein Low-Signal dargestellt wird. Außerdem handelt es sich bei Bild 23 um die Signalpegel auf dem Bus, d. h. also, daß alle Signalpegel außer Ready-for-Data und Data-

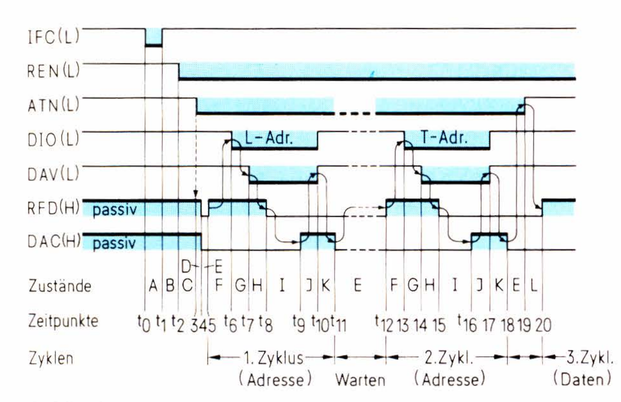


Bild 23. Start einer Datenübertragung über den IEC-Bus

Accepted Low-gehende Signale sind, d. h. sie beinhalten in Klammern ein „L"-Zeichen. *Diese von der IEC-Bus-Norm abweichende Schreibweise vermeidet ein dauerndes Umdenken bei Not-Ready-for-Data und Not-Data-Accepted!*

Bevor in einem solchen System etwas passieren kann, muß das Steuergerät die Geräte, die für die Datenübertragung aktiv werden sollen, adressieren. Das bedeutet, daß mindestens ein Gerät als Listener und nur ein Gerät als Talker adressiert werden muß; das geht so vor sich, daß als erstes das Steuergerät die Leitung ATN auf Low, d. h. in den aktiven Zustand legt. Alle angeschlossenen Interfaces hatten vorher ihren passiven Zustand, da ja die Interfaces alle im Ruhezustand waren, d. h. sämtliche Leitungen des IEC-Bus-Systems liegen im High-Zustand. Nachdem Attention (ATN) auf Low geht, melden sich alle Geräte innerhalb einer vorgegebenen Zeit mit einem Ready-for-Data-Signal (RFD) zurück. Außerdem müssen alle Geräte die Data-Accepted-Leitung (DAC) in den Null-Zustand legen, d. h. kein Gerät hat bisher Daten akzeptiert. Wenn diese Vorbedingung erfüllt ist, gibt das Steuergerät

Tabelle 2. Genormter Code für die Adressen

Lfd. Nr.	7-Bit-Code 765 4321	Zeichen	
0	000 0000	NUL	Addressed-command-group
.	.		
.	.		
.	.		Adressierte Befehle
15	000 1111	SI	
16	001 0000	DLE	Universal-command-group
.	.		
.	.		
.	.		
31	001 1111	US	Universal-Befehle
32	010 0000	SP	Listener-address-group
.	.		Listener-Adressen
.	.		
.	.		
62	011 1110	>	
63	011 1111	?	Unlisten-Adresse
64	100 0000	@	Talker-Address-group
.	.		
.	.		Talker-Adressen
94	101 1110	∩	
95	101 1111	—	Untalk-Adresse

auf den Datenleitungen z. B. eine Listener-Adresse aus, meldet sie über die Data-Valid-Leitung (DAV) gültig und wartet, daß das Gerät, welches diese ausgegebene Listener-Adresse benutzt, die Ready-for-Data-Leitung auf Null zieht. Nachdem das Gerät, welches die ausgesendete Adresse hat, diese empfangen hat, meldet es über die Data-Accepted-Leitung, daß es die Adresse empfangen hat, so daß das Steuergerät weiß, es kann die Adresse und das Data-Valid-Signal wieder wegnehmen.

Daraufhin verschwindet auch das Data-Accepted-Signal. Jetzt muß das Steuergerät wieder auf die Ready-for-Data-Leitung warten. Erst wenn der Listener, der gerade als solcher adressiert ist, bereit ist zur Übernahme neuer Daten, geht die Ready-for-Data-Leitung in den High-Zustand und das Steuergerät kann als nächstes eine Talker-Adresse aussenden; nach einer Einschwingzeit für die Datenleitungen meldet es diese Talker-Adresse über die Data-Valid-Leitung als gültig. Der Ablauf ist jetzt genau der gleiche wie vorher bei der Listener-Adresse und nach der Übertragung der Talker-Adresse geht der Bus wieder in den Ruhezustand, wie vorher schon zwischen der Übertragung der Listener- und der Talker-Adresse. Das adressierte Gerät muß die Tatsache, daß es adressiert worden ist, speichern. Jedes Gerät vergleicht die vom Steuergerät ausgesendeten Adressen mit der eigenen Adresse; diese ist einstellbar. Wenn der Vergleich positiv ausfällt, muß das Interface in einem Flipflop speichern, daß es adressiert worden ist.

Nachdem jetzt ein Gerät als Listener und ein Gerät als Talker adressiert ist, kann die ATN-Leitung in den Ruhezustand umgeschaltet werden und die Datenübertragung kann folgen. Die Datenübertragung sieht im Prinzip genauso aus, wie vorher die Übertragung der Adressen. Während allerdings die Adressen aus dem Steuergerät ausgesendet wurden, und dieses dazu auch das Data-Valid-Signal lieferte, kommen bei der Datenübertragung die Daten und das Data-Valid-Signal aus dem als Talker adressierten Gerät. Während vorher die Ready-for-Data- und die Data-Accepted-Leitung von dem jeweils zu adressierenden Gerät mit Signalen versorgt wurden, wird bei der Datenübertragung Ready-for-Data und Data-Accepted vom Empfänger ausgesendet.

In Bild 23 sind außer den Signalen IFC bis DAC noch weitere Zeilen eingeführt; in ihnen sind *Zustände, Zeitpunkte* und *Zyklen* eingezeichnet. Um bei der Festlegung der IEC-Bus-Schnittstelle eine möglichst „wasserdichte" Beschreibung zu finden, ist das gesamte Bus-System mit Zustandsdiagrammen beschrieben worden; darin werden die Bedingungen beschrieben, zu denen die jeweiligen Interfaces in den angeschlossenen Geräten von einem Zustand in den anderen übergehen. Wenn wir uns jetzt in Bild 23 während des ersten Datenübertragungszyklus den Zustand „F" ansehen, so ist dieser Zustand „F" der letzte Augenblick, bevor es zur Ausgabe von Daten aus dem Steuergerät kommt. Im Zustand „G" werden zwar schon Daten aus demselben ausgegeben, sie werden aber noch nicht gültig gemeldet. Im Zustand „H" folgt jetzt die Gültigmeldung, aber das zu adressierende Gerät hat noch nicht mit der Ready-for-Data-Leitung reagiert...und so geht es immer weiter.

Wenn man nicht die ruhenden Zustände zwischen den einzelnen Übergängen definieren will, so gibt es bestimmte Zeitpunkte, die in Bild 23 von $t_0...t_{20}$ numeriert sind. Zu den jeweiligen Zeitpunkten schaltet das Bus-System von einem Zustand in den anderen, z. B. wird zum Zeitpunkt t_7 der Übergang von „Daten nicht gültig" auf „Daten gültig" erfolgen. In der dritten Zeile sind die einzelnen Übertragungszyklen mit den dazwischenliegenden Wartezeiten eingetragen.

Codierung

Der Code für die Adressen ist genormt. Es handelt sich da bei um den ISO-7-Bit-Code; dieser ist bis auf geringe Abweichungen identisch mit dem ASCII-Code. Für die Übertragung der Daten besteht hingegen keine bindende Norm. Man kann also eine völlig codetransparente Datenübertragung vornehmen, d. h. jeder kann einen beliebigen Code benutzen. Natürlich ist dies kein idealer Zustand, aber die Norm läßt im Prinzip dem Anwender diese Freiheit. Die weitaus meisten Anwender werden auch für die Übertragung der Daten den gleichen ISO-7-Bit- bzw. ASCII-Code benutzen.

Tabelle 2 zeigt die genormte Adressen-Codierung. Mit einem 7-Bit-Code lassen sich 128 verschiedene Code-Kombinationen darstellen; diese sind in vier Hauptgruppen aufgeteilt; jede enthält 32 Kombinationen. Die ersten 32 Kombinationen sind *Befehlskombinationen*, davon sind die ersten 16 Kombinationen sogenannte „adressierte Befehlskombinationen" und die zweiten 16 sind sogenannte Universal-Befehlskombinationen.

Die zweite Gruppe von 32 Kombinationen sind *Listener-Adressenkombinationen*, d. h. 31 davon werden wirklich als Listener-Adressen verwendet, während die letzte Kombination in dieser Hauptgruppe das sogenannte „Unlisten"-Kommando ist. Mit dem Unlisten-Kommando lassen sich alle Geräte aus ihrem Listener-Zustand herausholen, d. h. jedes Gerät, das als Listener adressiert ist, wird durch das Unlisten-Kommando deadressiert.

Die dritte Hauptgruppe ist reserviert für *Talker- oder Senderadressen*, auch hier gibt es 31 Sendeadressen und eine Untalk- oder Nicht-Sendeadresse-Funktion wie bei Unlisten. Die vierte Hauptgruppe dient für noch komplexere *Erweiterungsmöglichkeiten*. Hier wollen wir nicht näher darauf eingehen.

Alle diese Befehle und Adressen werden mit *Attention* (ATN) zusammen übertragen. Immer wenn Attention nicht im aktiven Zustand ist, handelt es sich nicht um solche Befehle oder Adressen. Der Unterschied zwischen „adressierten Befehlen" und „Universal-Befehlen" liegt darin, daß adressierte Befehle nur auf adressierte Geräte wirken, d. h. diese Befehle gehen zwar an alle Geräte, wirksam werden sie aber nur, wenn das Gerät adressiert ist, während die *Universalkommandos* grundsätzlich an alle Geräte gehen und auch in allen Geräten ausgeführt werden, d. h. auch in denen, die nicht im adressierten Zustand sind.

Wenn man sich die Taelle 2 zusammen mit einer Code-Tabelle für den ASCII- oder ISO-7-Bit-Code ansieht, so stellt man fest, daß die Listener- und die Talker-Adressen, d. h. also die Adressen, die am meisten gebraucht werden, auch die gebräuchlichsten Zeichen aus der Code-Tabelle sind, d. h. Sonderzeichen, wie Ausrufe-Zeichen, Klammern, Sterne, Plus und Minus, die Ziffern und die großen Buchstaben. Die kleinen Buchstaben, die z. B. auf einer normalen Teletype-Fernschreibmaschine nicht erzeugt werden können, liegen in der vierten Hauptgruppe. Die erste Hauptgruppe besteht aus Sonderzeichen, die man mit einer Teletype-Fernschreibmaschine erzeugen kann.

Nachdem wir in Bild 23 gesehen haben, wie die Übertragung von Adressen und Daten erfolgt, nämlich in jedem Zyklus ein Zeichen, wollen wir uns in *Tabelle 3* einmal ein kleines Stückchen Datenübertragung über den IEC-Bus ansehen; denn ganz so einfach, wie es in Bild 23 dargestellt ist, ist die Datenübertragung leider doch nicht. In besagter Tabelle gibt es fortlaufend numerierte Zyklusnummern und zu jedem Zyklus eine Erläuterung, was als nächstes an Datenübertragung stattfindet. Als Beispiel ist hier die Übertragung eines Meßwertes aus einem Voltmeter zu einem Drucker gewählt. Das hier angegebene Beispiel ist nur eine von verschiedenen Möglichkeiten, wie eine solche Datenübertragung funktioniert. Wie es genau aussieht, kann man nur aus den Unterlagen des jeweiligen Gerätes entnehmen; denn jedes Gerät oder jeder Geräteentwickler hat innerhalb der vorgegebenen Norm noch gewisse Freiheiten, mit welchen Codes er welche Funktion veranlassen will.

Tabelle 3. Beispiel einer Datenübertragung über den IEC-Bus

Zyklus-Nr.	DIO-Nachricht DIO-Message	ATN	Bedeutung	DIO z. B.
1	L-Adresse	1	Adressieren Voltmeter	6
2	Programmier-daten	0	Bereich = Range	R
3		0	100 mV = 10^5 µV	5
4		0	Gleichspannung = DC	D
5		0	Start 1 Messung	S
6	UNL-Adresse	1	Listener-Adresse wegn.	?
7	L-Adresse	1	Adressieren Drucker	$
8	T-Adresse	1	Adressieren Voltmeter	V
Warten auf Ende der Messung				
9	Meß-daten	0	Datenblock	+/−
10		0	Datenblock	1
11		0	Datenblock	2
12		0	Datenblock	3
13		0	Datenblock	4
14	EOS-Zeichen	0	Ende-Zeichen	Cr/Lf
15	UNL-Adresse	1	Druckeradresse wegn.	?
16	UNT-Adresse	1	Voltmeteradresse wegn.	Unterstr.

Im ersten Zyklus der Tabelle 3 muß das Steuergerät das Voltmeter als Listener adressieren; diese Adressierung als Listener (!) ist deshalb erforderlich, weil das im Beispiel angenommene Voltmeter programmiert werden kann. Für die Übertragung dieser Listener-Adresse muß die Attention-Leitung auf aktiven Zustand gezogen werden, d. h. der Code der Attention-Leitung ist „1", dies bedeutet aber auch, daß auf dem Bus ein Low-Pegel übertragen wird.

Nachdem das Voltmeter als Listener adressiert ist, erfolgt in den nächsten Zyklen die *Programmierung des Voltmeters;* da es sich dabei um die Übertragung von Programmierdaten zum Voltmeter handelt, wird jetzt vorläufig die Attention-Leitung auf Null (High!) liegen. Als erstes wird der Bereich des Voltmeters programmiert und die Tatsache, daß es sich jetzt um eine Bereichsprogrammierung handelt durch die Übertragung des Buchstabens „R" (wie Range) übermittelt. Im 3. Zyklus wird der Bereich selber angegeben, und im Beispiel ist davon ausgegangen, daß eine „5" übertragen wird, um dem Voltmeter mitzuteilen, daß es im Meßbereich 10^5 µV ≙ 100 mV messen soll. Im 4. Zyklus wird dem Voltmeter die Meßart, nämlich „D" (wie DC) für Gleichspannung übermittelt. Im 5. Zyklus erfolgt die Übertragung eines „S" für den Startbefehl. Das Voltmeter beginnt daraufhin mit einer Messung. Im 6. Zyklus wird die Unlisten-Adresse übertragen, d. h. die Attention-Leitung geht auf den aktiven Zustand und das Voltmeter ist damit als Listener nicht mehr adressiert. In der letzten Spalte, wo der Code angegeben wird, der für die Datenübertragung auf dem DIO-Bus vorliegt, ist hierfür ein Fragezeichen angegeben; denn die Unlisten-Adresse ist ein Fragezeichen. Im 7. Zyklus

wird der Drucker als Listener adressiert. Hierfür ist ein Dollarzeichen vorgesehen. Im 8. Zyklus wird das Voltmeter wieder adressiert, diesmal allerdings als Talker.

Bei den letzten beiden Zyklen war die Attention-Leitung immer noch im aktiven Zustand, da es sich um die Übertragung von Adressen handelt. Um das Voltmeter als Talker zu adressieren, wird ein „V" für Voltmeter benutzt.

An dieser Stelle müssen wir noch eine kurze Erläuterung einfügen, warum die Listener-Adresse im Voltmeter eine „6" war. Wenn man für die Talker-Adresse des Voltmeters ein „V" benutzen möchte, dann ist es sehr sinnvoll, für die Listener-Adresse die Ziffer 6 zu benutzen, da das „V" und die „6" an den letzten 5 Stellen der Code-Kombination übereinstimmen! Aus dieser Überlegung kommt auch, daß der Drucker als Listener-Adresse ein Dollarzeichen hat, denn dieses stimmt in den letzten 5 Bit-Stellen mit dem großen „D" wie Drucker überein. Wenn der Drucker also noch zusätzlich eine Talker-Adresse hätte, dann könnte man ohne große Änderungen bei den Adressen zu erhalten, hierfür das „D" benutzen.

Zurück zur Tabelle 3: Zwischen dem 8. und dem 9. Zyklus muß jetzt, bevor die wirkliche Übertragung von Meßwerten aus dem Voltmeter zum Drucker erfolgen kann, darauf gewartet werden, daß das Voltmeter seine Messung beendet; dies kann im einfachsten Fall dadurch erfolgen, daß der Drucker zwar, nachdem er adressiert ist, bereit ist, Daten aufzunehmen, d. h. er meldet Ready-for-Data; aber aus dem Voltmeter gibt es keine Meldung Data-Valid. In einfachen Systemen, wo man während solcher Wartezeiten nichts anderes ausführen will, kann man die Wartezeit auf diese einfache Art und Weise überbrücken. In komplexeren Systemen wird während der Wartezeit noch diverse andere Arbeit auszuführen sein; in einem solchen Fall darf das Voltmeter nicht als Talker adressiert werden, sondern es muß sich über die zuvor schon beschriebene Service-Request-Leitung melden, wenn es mit seiner Messung fertig ist. Nachdem das Voltmeter in unserem Beispiel mit der Messung fertig ist, meldet es sich über die Data-Valid-Leitung und startet damit den 9. Zyklus.

Im 9. bis 13. Zyklus erfolgt nun automatisch ein sogenannter *Blocktransfer*, d. h. der gesamte Block von Daten aus dem Voltmeter wird in einem Zug, ohne daß das Steuergerät zwischendurch eingreifen muß, vom Voltmeter zum Drucker übertragen. Natürlich gibt es hierbei jetzt wieder den Hand-shake-Zyklus zwischen den beiden Geräten; aber das Steuergerät selber ist daran nicht aktiv beteiligt, denn das Voltmeter arbeitet als Talker und der Drucker als Listener.

Nachdem das gesamte Wort, d. h. also der Meßwert, aus dem Voltmeter zum Drucker übertragen ist, muß das Steuergerät irgendwie erkennen, daß der Block beendet ist. Eine Festlegung auf eine definierte Blocklänge wäre eine zu starke Einschränkung, und deshalb hat man an dieser Stelle ein *End-of-String-Zeichen* eingeführt. Dieses EOS-Zeichen kann z. B. ein Wagenrücklauf- oder ein Zeilenvorschub-Zeichen sein; dieses wird im 14. Zyklus nach dem eigentlichen Meßwert automatisch aus dem Voltmeter übertragen und anschließend hört es mit seiner Senderfunktion auf. Nachdem das Steuergerät das EOS-Zeichen auf dem Bus entdeckt hat, übernimmt es wieder aktiv die Befehlsangabe über den Bus und gibt in den nächsten beiden Zyklen die Unlisten- und die Untalk-Adressen aus. Damit sind Drucker und Voltmeter nicht mehr adressiert und das Steuergerät kann eine neue Arbeit beginnen.

Die beschriebene Methode dient zur einfachen Kennzeichnung eines Blockendes. Im Abschnitt 2 wurde bereits die zusätzliche Kennzeichnung des letzten Datenbytes durch die EOI-Leitung erläutert. Beide Ende-Meldungen sind in der Norm

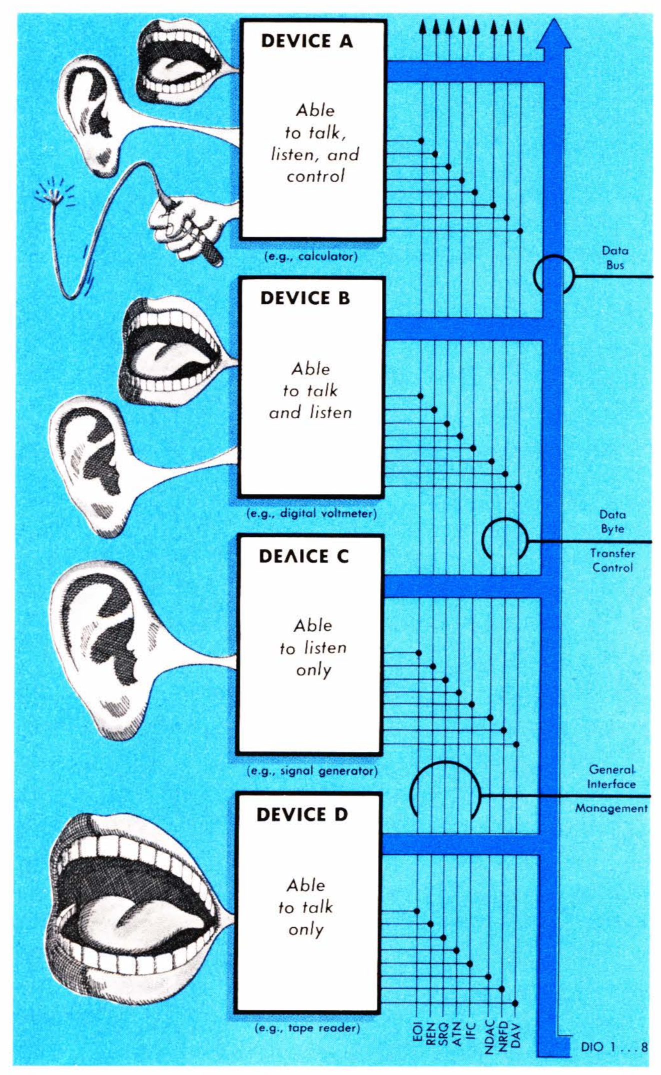


Bild 24. So zeigt die Firma Motorola in einer amerikanischen Anzeige den IEC-Bus – gewiß eine „sprechende" Darstellung! Die Peitsche symbolisiert die befehlende Tätigkeit des Steuergeräts

erlaubt. Die Aussendung eines besonderen Ende-Zeichens als extra Datenbyte hat den Vorteil, daß in Kleinstanlagen, wenn z. B. nur ein Voltmeter und ein Drucker zusammenarbeiten sollen, der Drucker so gebaut werden kann, daß er am Ende eines Datenblockes auf das End-of-String-Zeichen (EOS) hin eine komplette Zeile mit einem Meßwert druckt.

Dieses Beispiel klingt im ersten Moment komplizierter als es ist; denn ein einfaches Steuergerät kann die benötigten Buchstaben einem Lochstreifen entnehmen, d. h. man hat einen Lochstreifen oder eine Lochkarte, in der nur die angegebenen Ziffern, Buchstaben und Sonderzeichen aus der letzten Spalte DIO von Tabelle 3 eingedruckt sind, und gesteuert durch den Hand-shake gibt jetzt dieser Lochstreifenleser oder dieses Steuergerät Zeile für Zeile aus seinem Programm aus und steuert damit den gesamten IEC-Bus. Erst für sehr viel kompliziertere Aufgaben reichen reine Ablaufsteuerungen nicht mehr aus, und man benötigt dann eine *programmierbare Steuereinheit*, die Verzweigungen ermöglicht und im Zweifelsfall auch Berechnungen und Datenreduzierungen vornehmen kann. Zum Schluß *Bild 24*, eine buchstäblich „sprechende" Darstellung zu unserem Thema.

Dipl.-Ing. Tonio Frühauf

IEC-Bus-Ratgeber

1. Teil

Im Jahre 1975 drangen die ersten Informationen über den IEC-Bus an die Öffentlichkeit. Heute hat sich dieses System auf dem Gebiet der elektronischen Meßtechnik weltweit durchgesetzt. Mehrere hundert IEC-Bus-Geräte sind auf dem Markt und jeder Meßtechniker kommt früher oder später in die Lage, sich mit IEC-Bus-gesteuerten Automaten auseinanderzusetzen. Dementsprechend groß ist die Zahl der inzwischen über den IEC-Bus erschienenen Publikationen, so daß das Wissen über die theoretische Funktion des Bus-Systems in großem Maße vorhanden ist. Wie immer, benötigt man aber in der Praxis zusätzliche Informationen, die über das theoretische Grundlagenwissen hinausgehen. Nach mehrjähriger IEC-Bus-Praxis soll an dieser Stelle nun der Versuch gemacht werden, die Erfahrungen, die sich als bedeutsam herausgestellt haben, in übersichtlicher und praxisnaher Form darzustellen.

1 Was ist IEC-Bus-kompatibel?

Kompatibel bedeutet, daß einzelne Systemkomponenten zu einem gemeinsamen funktionsfähigen System kombiniert werden können, ohne daß spezielle Anpassungsmaßnahmen erforderlich sind. Bei IEC-Bus-Systemen ist sowohl die Bedingung der Hardware-Kompatibilität als auch die der Software-Kompatibilität zu erfüllen. Ein IEC-Bus-Gerät ist hardwarekompatibel, wenn zwei Bedingungen erfüllt sind:

1. Das Gerät muß mechanisch in bezug auf Anschlußstecker und Pinbelegung geeignet sein.
2. Das Gerät muß elektrisch, d. h. in seinen Interfacefunktionen und -Abläufen kompatibel sein.

Sind diese Bedingungen erfüllt, ist es grundsätzlich möglich, Daten zwischen entsprechenden Geräten auszutauschen. Diese Kompatibilität wird im wesentlichen durch Einhaltung der entsprechenden IEC-Bus-Norm gewährleistet; das heißt, Geräte, die konsequent nach der Norm ausgeführt sind, sind grundsätzlich mechanisch und elektrisch kompatibel.

Zusätzlich ist aber die Forderung nach Software-Kompatibilität zu beachten. Das heißt, sowohl der im jeweiligen Gerät verwendete Code, als auch die mit Hilfe dieses Codes realisierte Darstellungsform der zu übertragenden Informationen muß kompatibel sein mit den übrigen im System verwendeten Geräten. Diese Software-Kompatibilität ist in der IEC-Bus-Norm nicht vorgeschrieben. Um so wichtiger ist es, bei der Entscheidung über den Einsatz von IEC-Bus-Geräten diesen Punkt zu berücksichtigen. Empfehlungen über die Darstellungsform von übertragenen Informationen werden in der IEC-Bus-Empfehlung IEC 625-2 (1) enthalten sein, die sich z. Z. in Vorbereitung befindet.

2 Wie heißt der IEC-Bus richtig?

Im Laufe der Zeit haben sich unterschiedliche Namen für den IEC-Bus eingebürgert. Der Grund liegt darin, daß bis vor kurzem die international gültige IEC-Bus-Norm noch nicht verabschiedet war. Vielmehr bestanden nur die allerdings fast identischen nationalen Normvorschläge. So basiert die in Deutschland übliche Bezeichnung IEC-Bus auf der deutschen Norm DKE 66.22 (2). In den USA hat die gleiche Norm dagegen die Bezeichnung IEEE 488/75. Daneben hat sich die Bezeichnung GPIB (*General Purpose Interface Bus*) eingebürgert. Nach Verabschiedung der internationalen Norm erhält das entsprechende Normpapier die dann für alle Länder gültige Bezeichnung IEC 625-1. Daraus folgt die international gültige Bezeichnung IEC-625-Bus mit dem Kurzzeichen



Wichtig ist vor allem, daß all diese Namen das gleiche Bus-System kennzeichnen bzw. daß Geräte mit diesen Bezeichnungen kompatibel sind.

3 Der richtige IEC-Bus-Stecker

Leider hat die historische Entwicklung der IEC-Bus-Technik mit sich gebracht, daß für IEC-Bus-Geräte zwei Steckersysteme auf dem Markt sind (*Bild 1*):

Der 24polige Amphenolstecker und der 25polige Cannonstecker. Der Grund liegt darin, daß ursprünglich die US-Norm den 24poligen Stecker enthielt, während in den europäischen Normvorschlägen der 25polige Stecker enthalten war. In der endgültigen internationalen Norm wird dem 25poligen Stecker der

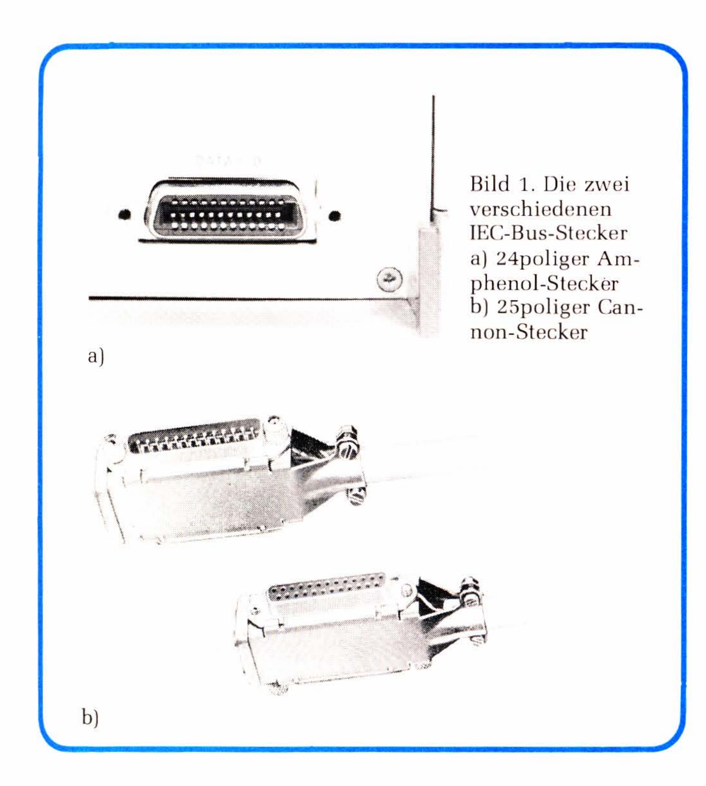


Bild 1. Die zwei verschiedenen IEC-Bus-Stecker
a) 24poliger Amphenol-Stecker
b) 25poliger Cannon-Stecker

Vorzug gegeben. In der Praxis ist aber heute die überwiegende Zahl der Geräte, insbesondere aus dem US-Bereich, mit 24poligen Steckern ausgerüstet. Vor der Zusammenstellung eines Systems ist deshalb zu klären, auf welchen Anschlußstecker man sich festlegen will. IEC-Bus-Kabel sind inzwischen für beide Steckertypen auf dem Markt verfügbar. Auch Übergangsstecker werden inzwischen angeboten, so daß eine gemischte Verwendung beider Steckertypen möglich ist.

Der 24polige Stecker bietet den Vorteil größerer Verbreitung, während der 25polige Stecker den Vorteil besserer HF-Dichtigkeit und sicherer EOI-Funktion bietet, da eine zusätzliche Masseleitung für die EOI-Funktionen zur Verfügung steht, die beim 24poligen Stecker fehlt. Allerdings wird die EOI-Leitung in der Praxis meist nur verwendet, wenn Daten nicht im ISO-7-Bit-Code übertragen werden und somit kein eindeutiges Schlußzeichen codierbar ist. Hier übernimmt die EOI-Leitung die Aufgabe, das Ende der Datenübertragung zu kennzeichnen.

4 Gesichtspunkte für die Auswahl von IEC-Bus-Geräten

Bereits bei der Konzeption eines IEC-Bus-Automaten ist die Entscheidung für die zu verwendenden Geräte zu treffen. Da diese Entscheidung unter Umständen auch den Kauf von Geräten bedeuten kann, ist es wichtig zu wissen, auf welche Punkte zu achten ist, um auch tatsächlich systemgeeignete Geräte zu erhalten. Neben der obengenannten Bedingung des richtigen Anschlußsteckers und der Einhaltung der IEC-Bus-Norm sind drei wesentliche Punkte zu beachten:

4.1 Die im Gerät realisierten IEC-Bus-Funktionen

In der IEC-Norm sind Bus-Funktionen enthalten, die in jedem Gerät realisiert sein müssen, um die Bedingung der Kompatibilität zu erfüllen. Man muß aber wissen, daß die IEC-Bus-Norm darüber hinaus zusätzliche Möglichkeiten und Modifikationen bietet, die als Optionen im Gerät enthalten sein können, um die Anwendungsmöglichkeiten zu erweitern, die aber für eine Kompatibilität nicht Bedingung sind. Es ist deshalb wichtig, daß der Anwender vor der Entscheidung für den Einsatz eines Gerätes sich ein genaues Bild über die im Gerät realisierten Funktionen macht.

Zu den grundsätzlich immer vorhandenen Funktionen gehört die Möglichkeit der Adressierung des Gerätes sowie die Eigenschaft des asynchronen Handshake-Betriebes. Es hängt jedoch vom Gerätetyp ab, ob Listener, Talker oder beide Funktionen realisiert sind und in welcher Form diese Funktionen realisiert sind. Zusätzliche Möglichkeiten, die im Gerät enthalten sein können, aber nicht müssen, sind Funktionen wie *Service Request, Serial Poll, Sekundäradressen* usw.

Alle diese Funktionen sind in der IEC-Bus-Norm durch Kurzzeichen gekennzeichnet. Die verschiede-

Programmierung (Option ZPV-B1)

System	IEC 625-1 (IEEE 488)
Anschluß	24polig, Amphenol
Schnittstellenfunktionen	
T6, TE 6	Sprechfunktion mit Sekundär-Adresse, Serienabfrage und autom. Entadressierung
L4	Hörerfunktion mit autom. Entadressierung
SR1	Bedienungsruffunktion (abschaltbar)
DC1	Rücksetzfunktion
DT1	Auslösefunktion
Zeitverhalten (typisch)	
Zeit für Adressierungsvorgang	1 μs
Zeit für Datenübernahme	0,5 ... 2 ms
Zeit zwischen Empfang der Sprecheradresse und Ausgabe des ersten Datenwortes	0,5 ms
Max. Datenausgabezeit/Zeichen	0,5 ms
Code	ISO-7-Bit
Zahlendarstellung	dezimal
Begrenzungszeichen	16 verschiedene Zeichen einstellbar (ab Werk ist CR eingestellt)

Bild 2. Beispiel für Datenangaben für ein Gerät mit IEC-Bus-Anschluß

nen Listener-Funktionen z. B. haben die Bezeichnungen L1...L4, die Talker-Funktionen die Bezeichnung T1...T8 usw. Die Kurzbezeichnungen sind (siehe auch IEC-Bus-Norm (2)) *Tabelle 1* zu entnehmen. Bei IEC-Bus-Geräten älteren Datums wird es oft schwierig sein, einen genauen Überblick über die realisierten Funktionen zu erhalten, da entsprechende Informationen nur mühsam aus dem Bedienungshandbuch herausgelesen werden können. Man geht deshalb zunehmend dazu über, die Datenblätter der Geräte um eine Auflistung der realisierten Interfacefunktionen zu erweitern. Dazu wird eine entsprechende IEC-Empfehlung erstellt werden. Auf diese Weise ist der Anwender in der Lage, auf einen Blick zu sehen, ob das Gerät die benötigten Funktionen enthält oder nicht (*Bild 2*).

4.2 Das Zeitverhalten des jeweiligen Gerätes

Für die erfolgreiche Zusammenstellung eines IEC-Bus-Systems ist es lebenswichtig, die notwendigen Informationen über das Zeitverhalten der einzelnen Geräte zu haben. Die Notwendigkeit dafür ergibt sich aus folgendem: Die Datenübertragung auf den IEC-Bus erfolgt asynchron. Das heißt, es wird keine feste Datenübertragungsrate verwendet. Vielmehr wird grundsätzlich auf das langsamste Gerät gewartet. Infolgedessen wird die Datenübertragungsgeschwindigkeit eines IEC-Bus-Systems durch das jeweils langsamste Gerät festgelegt. Um deshalb zu vermeiden, daß innerhalb eines Systems eine Vielzahl von hochwertigen, schnellen Geräten durch ein einziges langsames Gerät behindert wird, sind entsprechende Kenntnisse notwendig.

Zu unterscheiden sind dabei Geräteeigenschaften, die grundsätzlich immer die Bus-Geschwindigkeit beeinflussen und Eigenschaften, die nur im Falle der Aktivierung des jeweiligen Gerätes von Bedeutung sind. Die Erfahrung zeigt, daß vor allem die *Adressierungszeit* von Bedeutung ist, da sie bei jedem Adressierungsvorgang, auch wenn das betreffende Gerät nicht adressiert wird, in Erscheinung tritt. Die Adressierungszeit ist die Zeit, die ein Gerät benötigt, um eine ausgesandte Adresse zu erkennen und zu ent-

Tabelle 1.
Kurzbezeichnungen der möglichen IEC-Bus-Funktionen

1) Steuerfunktion

C1 Systemsteuerung
C2 IFC senden und in Einsatz gehen
C3 REN senden
C4 auf SRQ ansprechen

	Schnittstellen-nachrichten senden	Steuerung übernehmen	Steuerung übergeben	Steuerung zurückfordern	Parallelabfrage	Steuerung synchron übernehmen
C5	X	X	X	X	X	X
C6	X	X	X	X	X	0
C7	X	X	X	X	0	X
C8	X	X	X	X	0	0
C9	X	X	X	0	X	X
C10	X	X	X	0	X	0
C11	X	X	X	0	0	X
C12	X	X	X	0	0	0
C13	X	X	0	0	X	X
C14	X	X	0	0	X	0
C15	X	X	0	0	0	X
C16	X	X	0	0	0	0
C17	X	0	X	X	X	X
C18	X	0	X	X	X	0
C19	X	0	X	X	0	X
C20	X	0	X	X	0	0
C21	X	0	X	0	X	X
C22	X	0	X	0	X	0
C23	X	0	X	0	0	X
C23	X	0	X	0	0	0
C25	X	0	0	0	X	X
C26	X	0	0	0	X	0
C27	X	0	0	0	0	X
C28	X	0	0	0	0	0

2) Sprecherfunktion T (mit Adreßerweiterung TE)

	Grundausführung des Sprechers	Serienabfrage	Nur-sprechen-Betrieb	Automatisches Entadressieren
T1	X	X	X	0
T2	X	X	0	0
T3	X	0	X	0
T4	X	0	0	0
T5	X	X	X	X
T6	X	X	0	X
T7	X	0	X	X
T8	X	0	0	X

3) Hörerfunktion L (mit Adressenerweiterung LE)

	Grundausführung des Hörers	Nur-hören-Betrieb	Automatisches Entadressieren
L1	X	X	0
L2	X	0	0
L3	X	X	X
L4	X	0	X

4) SR1 – Bedienungsruffunktion (SRQ)
5) RL1 – Fern-/Eigen-Umschaltfunktion
RL2 – Fern-/Eigen-Umschaltfunktion ohne Blockierung der Eigenfunktion
6) PP1 – Parallel-Abfrage-Funktion
PP2 – Parallel-Abfrage nur durch geräteinterne Nachricht auslösbar
7) DC1 – Geräterücksetzfunktion
DC2 – Rücksetzfunktion ohne selektives Rücksetzen eines einzelnen Gerätes
8) DT1 – Gerätetriggerfunktion
9) SH1 – Sender-Handshakefunktion
10) AH1 – Empfänger-Handshakefunktion

scheiden, ob sie mit der eigenen Adresse übereinstimmt oder nicht. Erst nach diesem Vorgang kann das Gerät wieder „*Ready for data*“ melden. Daneben kann von Bedeutung sein, ob das jeweilige Gerät auch im nicht adressierten Zustand am Handshake-Zyklus teilnimmt und wenn ja, mit welcher Geschwindigkeit.

Während die bisher genannten Zeiten in jedem Falle die Gesamtgeschwindigkeit des Automaten beeinflussen, sind die Antwortzeiten im adressierten Zustand vor allem für die Leistungsfähigkeit des jeweiligen Gerätes im automatischen Betrieb von Interesse. Wesentlich sind hier vor allem die typischen Zeiten, die das Gerät entweder benötigt, um im Listener-Betrieb ein Datenwort zu empfangen, bzw. um im Talker-Betrieb ein Datenwort auszusenden (*Bild 3*).

Je nach Geräteentwicklung liegen die genannten Zeiten im Bereich von einigen 100 µs bis zu einigen ms, können in Sonderfällen aber auch Werte bis zu 100 ms und mehr erreichen, was dann die Verwendbarkeit in automatischen Testsystemen erheblich einschränkt. Wie bei der Angabe der IEC-Bus-Funktionen bemüht man sich auch bei den Zeiten, ausführlichere Angaben in den Datenblättern zu machen.

4.3 *Codierung und Syntax*

Während in der IEC-Bus-Norm die Hardware-Funktionen sehr exakt definiert wurden, besteht im Bereich der Codierung und Formatierung der zu übertragenden Informationen eine sehr große Freiheit. Infolgedessen ist es wichtig, vor dem Einsatz von Geräten Klarheit über die jeweils verwendeten Codezeichen und Darstellungsformen zu gewinnen.

4.3.1 Code

In den meisten Fällen wird bei IEC-Bus-Geräten der ISO-7-Bit-Code verwendet (siehe *Tabelle 2*). Da jedoch nicht alle Steuergeräte alle Zeichen aus dieser Tabelle ausgeben können, ist zu prüfen, ob das Gerät nur die üblicherweise verwendeten Großbuchstaben sowie Zahlen und Satzzeichen verwendet, oder ob zusätzliche Steuerzeichen aus anderen Spalten der ISO-7-Bit-Tabelle verwendet werden. Während man im ersten Falle mit großer Wahrscheinlichkeit keine Schwierigkeiten mit dem Steuergerät bekommt, wird man im zweiten Falle prüfen müssen, ob das verwendete Steuergerät in der Lage ist, die entsprechenden Sonderzeichen zu verarbeiten.

Da der ISO-7-Bit-Code keine optimale Datenpakkungsdichte ermöglicht, gibt es außerdem Fälle, in denen die Meßgeräte völlig andere Codierungen verwenden. Ein Beispiel ist die Verwendung von binären 8-Bit-Worten, die die optimale Ausnützung des Datenbusses ermöglichen. Dies tritt insbesondere bei Meßgeräten auf, die einen hohen Datendurchsatz haben. In diesem Falle können beliebige Bit-Kombinationen auftreten, so daß geprüft werden muß, ob das Steuergerät in der Lage ist, entsprechende Daten-Codierungen zu verarbeiten. (Beispiele sind Digitalvoltmeter mit hoher Meßrate oder auch Verbindungen zwischen Geräten mit Datenspeichern, die nicht nur Einzelmessungen, sondern vollständige Datensätze ausgeben müssen, wie dies z. B. bei Digitalisierungsgeräten der Fall ist (*Bild 4*)).

4.3.2 Syntax und Formatierung

Weiterhin ist zu prüfen, in welcher Form die Geräte zu übertragende Informationen darstellen. Das heißt, wenn z. B. der ISO-7-Bit-Code verwendet wird, so ist damit noch nicht festgelegt, in welcher Form z. B. Zahlen oder Dimensionen übertragen werden. Beispielsweise kann eine Zahl sowohl in Festkommadarstellung als auch in Exponenten-Darstellung übertragen werden. Um diesem Mangel abzuhelfen, wird die IEC-Bus-Norm durch einen Zusatz erweitert, der entsprechende *Empfehlungen bezüglich der Codierung und Formatierung* von Daten enthält. Da dieser Normvorschlag aber längere Zeit nach der Schaffung der Grundnorm entstand, existiert inzwischen eine Vielzahl von Geräten auf dem Markt, die nicht dieser Empfehlung folgen.

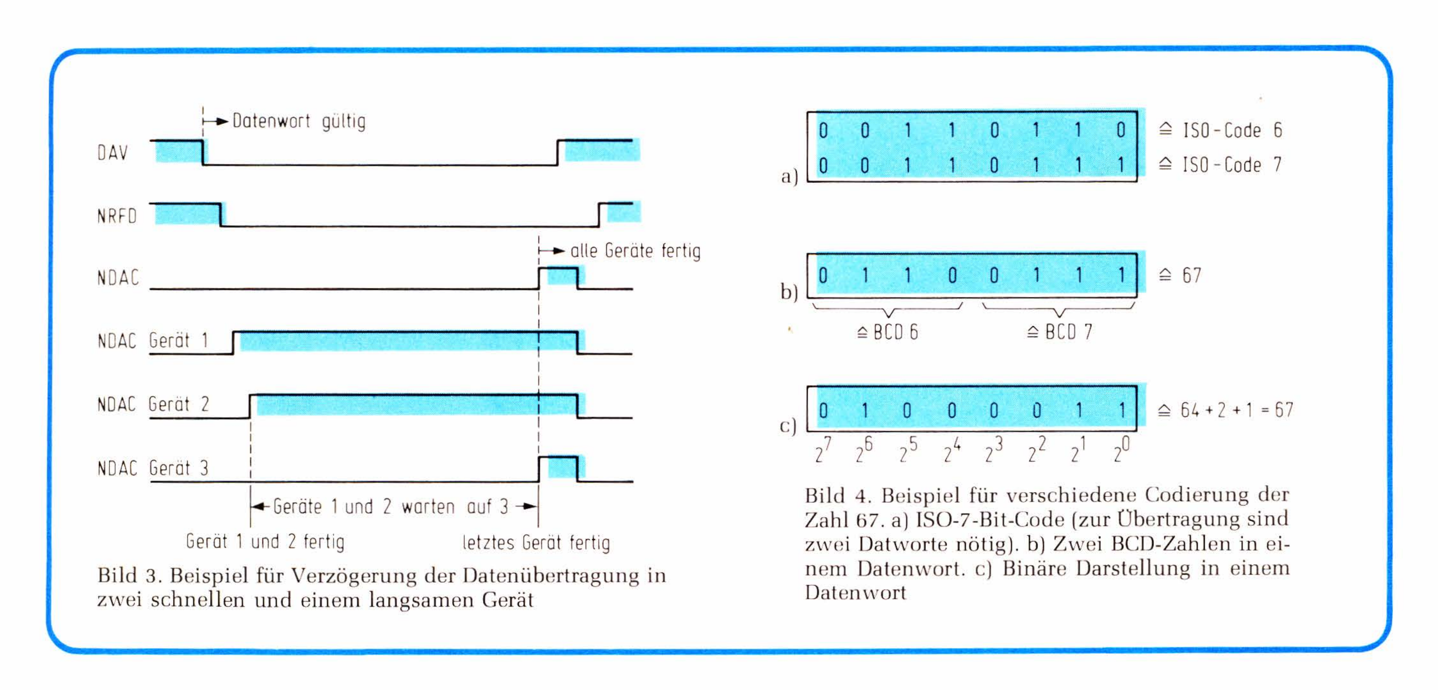


Bild 3. Beispiel für Verzögerung der Datenübertragung in zwei schnellen und einem langsamen Gerät

Bild 4. Beispiel für verschiedene Codierung der Zahl 67. a) ISO-7-Bit-Code (zur Übertragung sind zwei Datworte nötig). b) Zwei BCD-Zahlen in einem Datenwort. c) Binäre Darstellung in einem Datenwort



Bild 5. Beispiele für Prozeßsteuergeräte.
a) Processcontroller PCC mit IEC-Bus-Meßgeräten (Rohde & Schwarz)





b) ►
Tischcomputer 9825 S (Hewlett-Packard)

Von großem Vorteil ist z. B., wenn ein Meßgerät Daten einschließlich Vorzeichen und Stellwert in einer Formatierung ausgibt, in der das Steuergerät diese direkt als Daten weiterverarbeiten kann, ohne über aufwendige Zwischenverarbeitung das Format zu korrigieren. Viele Rechner sind z. B. in der Lage, die Darstellung +7.532 E+3 direkt als Zahl zu übernehmen. Gibt dagegen ein Voltmeter z. B. Wert der Digitalanzeige und Meßbereich getrennt aus, etwa in der Form 7.532 R3 (R3=Range3), so muß der Steuerrechner erst Bereich und Anzeige zum Meßwert zusammensetzen, was Programmspeicher und Rechenzeit kostet.

5 Die Auswahl des richtigen Steuergerätes

Wesentlicher Bestandteil eines IEC-Bus-Testsystemes ist das Steuergerät (*Controller*), das bisweilen aus einem Minicomputer, in den meisten Fällen aber aus einem Tischrechner besteht. Bei der Auswahl eines Rechners sind analog zu den Meßgeräten die drei Punkte, nämlich realisierte IEC-Bus-Funktionen, Geschwindigkeitsverhalten und Software, zu beachten.

Während Listener- und Talker-Funktion normalerweise in allen Rechnern realisiert sind, ist zu prüfen, ob die Funktionen „*Service Request, Serial-Poll* und evtl. *Paralell-Poll*" realisiert sind. Ohne diese Funktionen ergibt sich eine reduzierte Anwendungsmöglichkeit des Automaten. Außerdem ist darauf zu achten, ob der Rechner in der Lage ist, *Sekundäradressen* auszusenden. Ist dies nicht der Fall, ergeben sich in einigen Anwendungsfällen Schwierigkeiten, weil manche Meßgeräte nur umständlich oder gar nicht ohne Sekundäradressen gesteuert werden können.

Das Zeitverhalten eines IEC-Bus-Steuergerätes wird im wesentlichen durch die *Rechengeschwindigkeit* bestimmt. Die Praxis zeigt, daß Aufbereitung der Daten und Adressierung vergleichsweise erheblich mehr Zeit benötigen, als die eigentliche Übertragung der Daten. Während diese sich im Bereich von z. B. 100 µs abspielt, wird für die Aufbereitung eine Zeit von einigen ms bis zu Werten von 10...20 ms (bei Tischrechnern) benötigt.

Wichtig sind die Möglichkeiten, die der Rechner besitzt, um den IEC-Bus softwaremäßig zu aktivieren. Üblicherweise besitzen IEC-Bus-kompatible Rechner je einen speziellen IEC-Bus-Befehl für Input und Output, mit denen direkt ISO-7-Bit-Zeichen verarbeitet werden können (z. B. PRINT und INPUT). Bei Bedarf ist aber zu prüfen, ob der Rechner zusätzlich die Möglichkeit besitzt, beliebige frei codierte Daten zu senden und zu empfangen. Dafür sind *spezielle Ein- und Ausgabebefehle* erforderlich, die Bestandteil des Rechnerbetriebssystems sein müssen (z. B. WYTE oder RBYTE). Nur in diesem Falle ist es normalerweise auch möglich, spezielle, standardisierte IEC-Bus-Befehle, wie z. B. „*Selected Device Clear*" oder „*Local lock Out*" zu übertragen.

Literatur

[1] Entwurf DIN IEC625 Teil 2. Okt. 79, Beuth Verlag GmbH, Berlin.
[2] Entwurf DIN IEC66. 22. Jan. 76, Beuth Verlag GmbH, Berlin.
Die Redaktion empfiehlt zum Nachschlagen in der ELEKTRONIK:
[3] *Klaus, J.*: Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.
[4] *Klein, P., Wilhelmy, H.*: Der IEC-Bus, ELEKTRONIK 1977, H. 10, S. 63...74.
[5] *Klein, M.*: Ein einfacher IEC-Bus-Monitor. ELEKTRONIK 1980, H. 2, S. 74...75.



Dipl.-Ing. Tonio Frühauf stammt aus Melle bei Osnabrück. Nach seinem Studium in Darmstadt und München trat er 1967 in die Firma Rohde & Schwarz ein, wo er jetzt als Abteilungsleiter in der elektronischen Meßtechnik (Entwicklung) tätig ist. Innerhalb des DKE und der Wg3 des TC66 der Internationalen IEC Normengruppe wirkt er auch bei der Normung des IEC-Bus mit.
Hobbys: Musik, Sport
ELEKTRONIK-Leser seit 1967

Dipl.-Ing. Tonio Frühauf

IEC-Bus-Ratgeber

2. Teil

Der erste Teil dieser Arbeit zeigte die wichtigsten grundlegenden Gesichtspunkte des IEC-Busses und der daran angeschlossenen Geräte. Der vorliegende Teil gibt Ratschläge für den Betrieb des IEC-Busses in der Praxis.

6 Die Inbetriebnahme von IEC-Bus-Automaten

Wenn nach den oben angeführten Gesichtspunkten alle Geräte eines Automaten sorgfältig ausgewählt wurden, wird normalerweise in kürzester Zeit eine erfolgreiche Funktion des Systems eintreten. Es soll hier jedoch auf einige in der Praxis häufig auftretende Schwierigkeiten hingewiesen werden.

6.1 *Das Verbinden der IEC-Bus-Anschlüsse des Systems*

6.1.1 Menge der angeschlossenen Geräte

Nachdem zunächst die meßtechnische Verkabelung des Automaten erfolgt ist, wird bei manuellem Betrieb die meßtechnische Funktionsfähigkeit des Systems überprüft. Anschließend erfolgt die Verbindung der IEC-Bus-Anschlüsse mit dem Steuergerät. Dabei müssen einige Gesichtspunkte sorgfältig beachtet werden: Der IEC-Bus ist für eine Zahl von max. 15 Geräten vorgesehen. Die Erfahrung zeigt aber, daß diese Grenze nur bei Verwendung von vollständig der IEC-Norm entsprechenden Geräten erreicht werden kann. Da dies erfahrungsgemäß nicht immer der Fall ist, können sich manchmal schon bei Kombinationen von mehr als zehn Geräten Schwierigkeiten einstellen. In diesem Falle sind die Regeln bezüglich der Länge der Verkabelung besonders streng zu beachten.

6.1.2 Länge der IEC-Bus-Kabel

Der IEC-Bus ist für eine maximale Länge von 20 Metern ausgelegt. Trotzdem sollte die Gesamtlänge der Kabel aus Gründen der Übertragungssicherheit möglichst den Wert von 15 Metern nicht überschreiten. Bei Verwendung der üblichen „Huckepack"-Stecker-Kabel gilt außerdem, daß die Verbindung von Gerät zu Gerät nicht länger als zwei Meter sein sollte. Generell sollten die Verbindungen von Gerät zu Gerät möglichst kurz gehalten werden. Es befinden sich dazu Kabel mit unterschiedlicher Länge bis herunter zu 0,5 m auf dem Markt. Da der IEC-Bus ein Datenübertragungs-System hoher Geschwindigkeit ist, ist außerdem darauf zu achten, daß kein am Ende leerlaufendes Kabel an das System angeschlossen ist (*Bild 6*).

Falls trotzdem Probleme in der Datenübertragung auftreten, sind diese im allgemeinen dadurch zu beheben, daß die Reihenfolge der angesteuerten Geräte verändert wird, und daß die Verbindung des Kabels vom Rechner zum System an unterschiedlichen Stellen angebracht wird. Fehler in der IEC-Bus-Datenübertragung äußern sich meistens darin, daß Daten verstümmelt oder unvollständig am Empfangsort ankommen. Falls trotz unterschiedlicher Kabelkonfigurationen keine Beseitigung der Fehler möglich ist, empfiehlt es sich, jeweils ein Gerät vom Bus zu trennen, um festzustellen, durch welches Gerät die Störung hervorgerufen wird. Häufig ist dann innerhalb des Gerätes nicht korrekte normgerechte Beschaltung der IEC-Bus-Leitungen mit Abschlußwiderständen oder nicht Hf-mäßige Masseverbindung der Masseleitungen die Ursache.

6.2 *Adressierung*

6.2.1 Einstellen der Geräteadressen an den Meßgeräten

Vor Inbetriebnahme des Systems müssen die Geräteadressen so eingestellt werden, daß jedes Gerät und jede Funktion durch eine individuelle Adresse aufgerufen werden kann. Das heißt, es ist darauf zu achten, daß keine Adresse zweimal eingestellt wird. In der Praxis hat es sich eingebürgert, daß bei Meßgeräten Listener- und Talkeradresse mit dem gleichen Schalter eingestellt werden. Infolgedessen können beide Adressen nur gemeinsam umgeschaltet werden. Im allgemeinen bedeutet dies, daß jeweils sich entsprechende Zeichen der ISO-7-Bit-Tabelle für Listener- und Talkeradresse verwendet werden. Dies ist aber von der IEC-Bus-Norm her keine zwingende Notwendigkeit, sondern ist nur in der Einsparung eines zweiten Adreßschalters begründet. Soweit notwendig und möglich können auch nicht korrespondierende Liste-

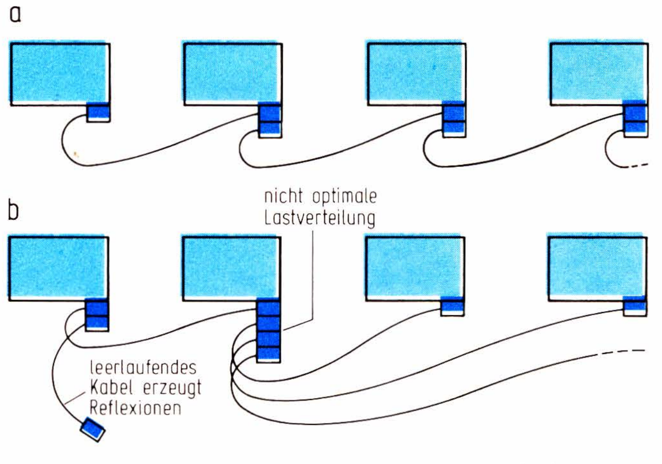


Bild 6. Verkabelung des IEC-Busses a) richtig, b) falsch

ner- und Talkeradressen eingestellt werden. Die Adresse kann durch die Benennung des entsprechenden ISO-7-Bit-Zeichens ausgedrückt werden. In der Praxis hat es sich aber eingebürgert, die in der Norm vorgesehenen Adressen von 1...31 durchzunumerieren (siehe *Tabelle 3*), während der Nummer 32 jeweils die Löschfunktion (Unlisten und Untalk) zugeordnet ist. Es gibt also max. 31 Listener- und 31 Talker-Adressen. Es empfiehlt sich, nach der Einstellung der Adressen, diese in irgend einer Form an den Geräten sichtbar zu markieren, um späteren Fehlern vorzubeugen. Die Art, in der die Adresse bei den einzelnen Geräten eingestellt wird, ist dem jeweiligen Handbuch zu entnehmen. Meist befindet sich der Adreßschalter an der Geräterückseite.

6.2.2 Festlegung der Geräteadresse beim Steuerrechner

Die Adressierung der jeweiligen Meßgeräte erfolgt im allgemeinen durch den gleichen Programmbefehl, durch den die auszugebende oder einzulesenden Zeichen definiert werden. Man muß sich dabei zunächst informieren, auf welche Weise die Adressen bei dem jeweils verwendeten Rechner gekennzeichnet sind. Im allgemeinen geschieht dies gemäß Tabelle 3 mit der Numerierung von 1...30. Je nachdem, ob ein Ausgabebefehl oder ein Einlesebefehl mit dieser Adresse kombiniert wird, sendet der Rechner die entsprechende Listener- oder Talkeradresse aus:
PRINT @ "XYZ" bedeutet also Listenadresse 26,
INPUT @ 26: A$ bedeutet also Talkeradresse 26.

Da die „Unlisten"- und „Untalk"-Befehle standardisiert sind, werden sie normalerweise nicht mit angegeben. Vielmehr löschen die meisten Rechner nach erfolgter Ausführung des jeweiligen Befehls automatisch wieder die vorher gesetzten Adressen.

Neben dieser besonders einfachen Art der Adressierung gibt es aber auch Rechner, die das anzusprechende Gerät als Datenfile betrachten. In diesem Falle wird zunächst das zum Gerät gehörende File eröffnet. Gleichzeitig wird dabei die IEC-Bus-Adresse des Gerätes dem File zugeordnet:
OPEN 3, 26
bedeutet beispielsweise, es wird das File Nummer drei eröffnet, dem die IEC-Bus-Adresse 26 zugeordnet ist. Bei jeder Ausgabe genügt es nun, das File 3 aufzurufen und die Adresse 26 wird automatisch ausgesandt z. B. PRINT # 3, "XYZ".

6.3 Inbetriebnahme der Listenerfunktionen

6.3.1 Funktionskontrolle (*Bild 7*)

Es empfiehlt sich, zunächst die Listener-Funktion der Geräte zu testen. Dazu wird vom Tischrechner aus ein Gerät als Listener adressiert und ein Programmierbefehl dorthin übertragen. Dies kann im allgemeinen mit ein oder zwei Programmzeilen geschehen.
Beispiel: PRINT @ 26: "XYZ".
Wenn der entsprechende Befehl beim Gerät ankommt, und der Rechner wieder bereit zeigt, bedeutet dieses, daß das gesamte Handshake-System des Automaten

Tabelle 2: ISO-7-Bit-Code

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	TC_7 (DLF)	SP	0	@	P	`	p
0	0	0	1	1	TC_1 (SOH)	DC_1	!	1	A	Q	a	q
0	0	1	0	2	TC_2 (STX)	DC_2	"	2	B	R	b	r
0	0	1	1	3	TC_3 (ETX)	DC_3	#	3	C	S	c	s
0	1	0	0	4	TC_4 (EOT)	DC_4	$	4	D	T	d	t
0	1	0	1	5	TC_5 (ENQ)	TC_8 (NAK)	%	5	E	U	e	u
0	1	1	0	6	TC_6 (ACK)	TC_9 (SYN)	&	6	F	V	f	v
0	1	1	1	7	BEL	TC_{10} (ETB)	'	7	G	W	g	w
1	0	0	0	8	FE_0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	FE_1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	10	FE_2 (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	11	FE_3 (VT)	ESC	+	;	K	[	k	{
1	1	0	0	12	FE_4 (FF)	IS_4 (FS)	,	<	L	\	l	|
1	1	0	1	13	FE_5 (CR)	IS_3 (GS)	–	=	M	]	m	}
1	1	1	0	14	SO	IS_2 (PS)	.	>	N	^	n	‾
1	1	1	1	15	SI	IS_1 (US)	/	?	O	_	o	DEL

Tabelle 3: Zulässige IEC-Bus-Adressen

	Listener-Adresse			Talker-Adresse		
Geräte-Nummer	ISO-7-bit	Dezimal-Wert	DIO-Bus	ISO-7-bit	Dezimal-Wert	DIO-Bus
0	SP	32	0 0 1 0 0 0 0 0	a	64	0 1 0 0 0 0 0 0
1	!	33	0 0 1 0 0 0 0 1	A	65	0 1 0 0 0 0 0 1
2	"	34	0 0 1 0 0 0 1 0	B	66	0 1 0 0 0 0 1 0
3	#	35	0 0 1 0 0 0 1 1	C	67	0 1 0 0 0 0 1 1
4	$	36	0 0 1 0 0 1 0 0	D	68	0 1 0 0 0 1 0 0
5	%	37	0 0 1 0 0 1 0 1	E	69	0 1 0 0 0 1 0 1
6	&	38	0 0 1 0 0 1 1 0	F	70	0 1 0 0 0 1 1 0
7	'	39	0 0 1 0 0 1 1 1	G	71	0 1 0 0 0 1 1 1
8	(	40	0 0 1 0 1 0 0 0	H	72	0 1 0 0 1 0 0 0
9	)	41	0 0 1 0 1 0 0 1	I	73	0 1 0 0 1 0 0 1
10	*	42	0 0 1 0 1 0 1 0	J	74	0 1 0 0 1 0 1 0
11	+	43	0 0 1 0 1 0 1 1	K	75	0 1 0 0 1 0 1 1
12	,	44	0 0 1 0 1 1 0 0	L	76	0 1 0 0 1 1 0 0
13	-	45	0 0 1 0 1 1 0 1	M	77	0 1 0 0 1 1 0 1
14	.	46	0 0 1 0 1 1 1 0	N	78	0 1 0 0 1 1 1 0
15	/	47	0 0 1 0 1 1 1 1	O	79	0 1 0 0 1 1 1 1
16	0	48	0 0 1 1 0 0 0 0	P	80	0 1 0 1 0 0 0 0
17	1	49	0 0 1 1 0 0 0 1	Q	81	0 1 0 1 0 0 0 1
18	2	50	0 0 1 1 0 0 1 0	R	82	0 1 0 1 0 0 1 0
19	3	51	0 0 1 1 0 0 1 1	S	83	0 1 0 1 0 0 1 1
20	4	52	0 0 1 1 0 1 0 0	T	84	0 1 0 1 0 1 0 0
21	5	53	0 0 1 1 0 1 0 1	U	85	0 1 0 1 0 1 0 1
22	6	54	0 0 1 1 0 1 1 0	V	86	0 1 0 1 0 1 1 0
23	7	55	0 0 1 1 0 1 1 1	W	87	0 1 0 1 0 1 1 1
24	8	56	0 0 1 1 1 0 0 0	X	88	0 1 0 1 1 0 0 0
25	9	57	0 0 1 1 1 0 0 1	Y	89	0 1 0 1 1 0 0 1
26	:	58	0 0 1 1 1 0 1 0	Z	90	0 1 0 1 1 0 1 0
27	;	59	0 0 1 1 1 0 1 1	[	91	0 1 0 1 1 0 1 1
28	<	60	0 0 1 1 1 1 0 0	\	92	0 1 0 1 1 1 0 0
29	=	61	0 0 1 1 1 1 0 1	]	93	0 1 0 1 1 1 0 1
30	>	62	0 0 1 1 1 1 1 0	^	94	0 1 0 1 1 1 1 0
		63	0 0 1 1 1 1 1 1	_	95	0 1 0 1 1 1 1 1

einwandfrei arbeitet, und man kann dazu übergehen, die Listener-Funktionen der übrigen Geräte zu testen. Ist dieses nicht der Fall, das heißt, der Befehl wird nicht ausgeführt oder er wird zwar ausgeführt, aber der Rechner wird nicht wieder bereit zur Ausführung des nächsten Programmes, so bedeutet dieses, daß der Handshake-Zyklus nicht einwandfrei funktioniert. Das heißt, eines der Geräte ist nicht in der Lage, dem Datenübertragungs-Zyklus zu folgen und blockiert deshalb den Bus. Wenn man ganz sicher sein will, kann man das nachprüfen: Am Ende der Datenübertragung muß die Leitung NRFD wieder auf +5 V liegen. Die Ursache kann jedes Gerät sein und nicht nur das jeweils in dem Programmbefehl angesprochene Gerät!

6.3.2 Ursachen für fehlerhafte Listenerfunktion

Häufige Fehler in der Praxis sind z. B. folgende: Bei einem oder mehreren Geräten des Systems wurde noch nicht die Netzspannung eingeschaltet. Oder bei Geräten mit nicht automatischer Umschaltung auf Programmierbetrieb bei Adressierung wurde versäumt, diese von Handbedienung auf Fernsteuerung umzuschalten. Dies kann je nach Gerät sowohl durch einen manuellen Schalter als auch durch Ansteuerung der REN-Leitung geschehen. In diesem Falle muß der Tischrechner die Möglichkeit besitzen, die REN-Leitung entsprechend zu aktivieren. Ist dies nicht der Fall, kann notfalls eine direkte Masseverbindung mit der REN-Leitung Abhilfe bringen. Man kann die Funktion der REN-Leitung leicht überprüfen: Sie muß, solange Daten übertragen werden, grundsätzlich auf 0 V liegen.

Sind alle diese Punkte beachtet, so kann bei trotzdem auftretender Fehlfunktion nur ein defektes Gerät oder Buskabel die Ursache sein. Dieses ist einfach zu lokalisieren, indem man der Reihe nach die einzelnen Geräte vom Bus abtrennt und erneut versucht, den Listenerbefehl zu übertragen.

6.4 Inbetriebnahme der Talkerfunktionen (Bild 8)

Um die Talkerfunktion eines Meßgerätes auszulösen, muß man zunächst die Voraussetzungen kennen, unter denen das Gerät bereit ist, Daten auszusenden. Im einfachsten Falle genügt das Aufrufen der Talkeradresse, um den Datenausgabevorgang auszulösen. Bei Geräten, die mehr als nur eine Größe ausgeben können, ist es dazu aber zunächst erforderlich, zu definieren, welche Größe ausgegeben werden soll. Dies kann durch einen dem Talkervorgang vorgeschalteten Listenervorgang geschehen, indem man dem Gerät die entsprechenden Informationen gibt. Bei manchen Geräten ist ohne diese vorhergehende Information trotz Setzen der Talkeradresse gar keine Datenausgabe möglich.

Eine andere Methode, die Datenausgabe zu differenzieren, ist die der Sekundäradresse. Dabei wird der eigentlichen Talkeradresse noch eine zweite Unteradresse nachgeschaltet, die die auszugebende Größe definiert

INPUT @ 26,3: A $ heißt beispielsweise:
Vom Gerät 26 soll die Größe Nr. 3 ausgegeben werden.

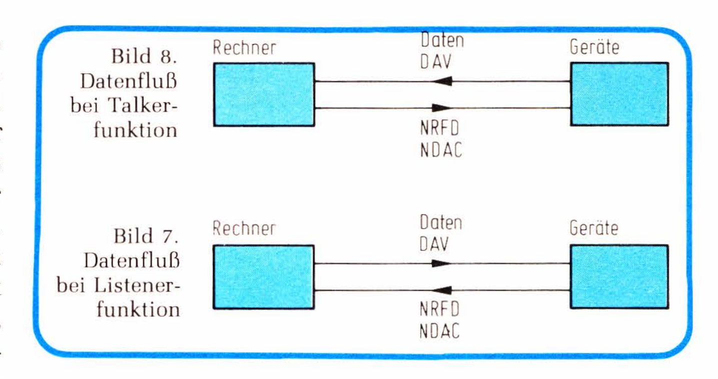


Bild 8. Datenfluß bei Talkerfunktion

Bild 7. Datenfluß bei Listenerfunktion

Allerdings besitzen nicht alle Rechner die Möglichkeit, diese Sekundäradressen auszugeben. Gute Meßgeräte besitzen deshalb niemals nur diese Ansteuerungsart, vielmehr kann man alternativ auch einen vorgeschalteten Listenerbefehl verwenden.

Während die Listenerfunktion in der Praxis fast immer im ersten Anlauf funktioniert, ergeben sich bei Inbetriebnahme der Talkerfunktion relativ häufig Schwierigkeiten. Dies liegt vor allem an dem nicht genormten Schlußzeichen.

6.4.1 Schlußzeichen

Da der Steuerrechner wissen muß, wann das jeweilige datenaussendende Gerät alle Daten ausgegeben hat, ist es erforderlich, zwischen Steuerrechner und Meßgerät ein Code-Zeichen zu vereinbaren, das grundsätzlich die Bedeutung des letzten übertragenen Zeichens hat. Dieser „Delimiter" ist nicht einheitlich festgelegt. In der Praxis werden je nach Gerätehersteller hauptsächlich die Zeichen CR und LF verwendet. Man muß also in jedem Falle darauf achten, daß das vom Meßgerät ausgesandte Schlußzeichen das gleiche ist, das der Rechner als Schlußzeichen benötigt.

Bei einigen Rechnern besteht dazu die Möglichkeit, zwischen den Schlußzeichen CR und LF zu wechseln oder auch andere Schlußzeichen zu definieren. Im Sinne einer einheitlichen Software ist es aber wünschenswert, im gesamten Testautomaten mit einem einheitlichen Schlußzeichen zu arbeiten. Aus diesem Grunde besitzen die meisten Meßgeräte mit Talkerfunktion die Möglichkeit, das Schlußzeichen umzuschalten. Diese Umschaltung erfolgt im allgemeinen im Gerät auf der IEC-Bus-Interface-Baugruppe. Vor Inbetriebnahme des Automaten sollte man deshalb möglichst ein für alle Geräte einheitliches Schlußzeichen festlegen, auf das dann alle Geräte soweit möglich eingestellt werden. Im allgemeinen wird dieses eines der beiden Zeichen CR oder LF sein.

Hinzu kommt bei manchen Geräten die Möglichkeit, gleichzeitig zum Schlußzeichen die Leitung EOI zu aktivieren (*Bild 9*). Dies bedeutet eine Komplizierung des Schlußzeichens, da zum einen der Rechner die Funktion EOI besitzen muß, und zum anderen in den meisten Fällen Systeme entstehen werden, bei denen einige Geräte die Funktion EOI verwenden und andere wiederum nicht. Trotzdem ist die Verwendung der Leitung EOI in einigen Fällen unumgänglich, nämlich dann, wenn Daten nicht im ISO-7-Bit-Code, sondern z. B. binär codiert übertragen werden, so daß die Vereinbarung eines Schlußzeichens nicht möglich

ist, weil im 8-Bit-Binärcode jede Zeichenkombination als Datenzeichen vorkommen kann, und die Reservierung eines bestimmten Zeichens nur als Schlußzeichen nicht möglich ist. Wird ein derartiges Gerät vom Tischrechner als Talker aufgerufen, so muß im Programm an der entsprechenden Stelle berücksichtigt werden, daß bei diesem Gerät das Schlußzeichen durch EOI gekennzeichnet wird. Das heißt, eine vereinbarte Bit-Kombination wird erst dadurch zum Schlußzeichen, daß gleichzeitig die EOI-Leitung gesetzt ist.

Wenn diese Punkte beachtet werden, kann man im allgemeinen davon ausgehen, daß auch der Talker-Datenverkehr einwandfrei funktioniert.

6.4.2 Kontrolle der Talkerdatenausgabe

Es empfiehlt sich, den vom Meßgerät ausgegebenen Wert zunächst als String einzulesen und auszudrukken, um die gesamte vom Meßgerät ausgegebene Funktion erkennen zu können. Grundsätzlich ist häufig auch das direkte Übernehmen einer vom Meßgerät ausgegebenen Größe in den Rechner in Form einer numerischen Variablen möglich. In diesem Falle werden vom Rechner jedoch unter Umständen Zusatzinformationen unterdrückt, die zunächst für das Austesten der Systeme interessant sein können.

```
Beispiel: INPUT @ 26: A $
          PRINT   A$      ergibt U + 1.07 E+1V
          INPUT @ 26: A
          PRINT   A       ergibt 10,7
```

Falls das System nach Übernahme des vom Meßgerät ausgegebenen Datenwertes nicht weiter funktionsfähig ist, kann die Ursache darin liegen, daß das Meßgerät noch weitere Daten ausgeben möchte, um erst dann seine Talkerfunktion zu beenden (s. auch 6.4.3). Es ist also wichtig zu wissen, daß ein Talkergerät erst dann sein Schlußzeichen sendet, wenn es alle auszugebenden Informationen ausgesandt hat, und daß erst das Schlußzeichen den Weiterlauf des Programmes ermöglicht. Im Störungsfalle muß bei Senden der Befehle IFC (*Interface Clear*) oder SDC (*Selected device Clear*) der Talkerzyklus vorzeitig beendet werden, und damit das System wieder frei werden.

6.4.3 Ausgabe von Datenfolgen

Im einfachsten Fall gibt ein Gerät nur einen einzigen Zahlenwert aus, der vom Rechner in Form einer einzigen Variablen gespeichert wird. Bei Anfall größerer Datenmengen sind manche Meßgeräte aber auch in der Lage, nach Empfang der Talkeradresse gleich mehrere Zahlen auszusenden. In diesem Falle müssen im Einlesebefehl genügend viele Variablen definiert werden, um alle Daten zu speichern, die das Gerät ausgeben will. Im Falle eines Phasenvoltmeters, das Betrag und Phase mißt, bedeutet dies z. B.

```
INPUT @ 26: B, P
```

In B ist dann der Betrag und in P die Phase enthalten. Auf diese Weise wird für beide Daten nur ein einziger Adressierungsvorgang benötigt, was die Datenübertragungsgeschwindigkeit erhöht. Wird in einem solchen Falle nur eine Variable definiert, so kann die zweite Größe nicht eingelesen werden, und das System ist blockiert.

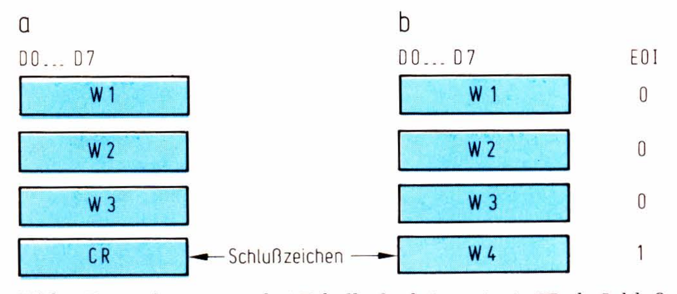


Bild 9. Datenübertragung bei Tabellarfunktion: a) mit CR als Schlußzeichen (auch LF möglich), b) mit EOI-Leitung und beliebigem Schlußzeichen

6.4.4 Ursachen für fehlerhafte Talkerfunktion

In der Praxis ist die häufigste Fehlerursache die Verwendung von falschen Schlußzeichen (siehe 6.4.1) und ganz allgemein das Nichtübereinstimmen vom Ausgabeformat der vom Gerät ausgegebenen Daten und dem entsprechenden Einleseformat des jeweiligen Steuerrechners (siehe 6.4.3).

Eine weitere Ursache kann die fehlende Entadressierung eines Gerätes sein: Meist war das Gerät, bevor es als Talker aufgerufen wurde, als Listener adressiert (*Bild 10*). Wenn das Gerät die Eigenschaft der automatischen Entadressierung hat, so wird automatisch beim Setzen einer Talkeradresse die Listeneradresse gelöscht. Ist dies aber nicht der Fall, so muß der Steuerrechner zunächst den Unlistenbefehl senden. Manche, aber nicht alle Rechner, tun dies automatisch nach jeder Datenausgabe, da diese Funktion zwar die Programmierung vereinfacht, aber andererseits nicht zeitoptimal ist. Ist die nicht durchgeführte Löschung der Listeneradresse die Ursache für blockierte Talkerfunktion, so ist dies an den bei den meisten Geräten vorhandenen Adreßkontrollampen leicht zu erkennen. Daneben kommen alle Ursachen in Frage, die auch für eine Fehlfunktion des Listenervorganges gelten (siehe 6.3.2). Darüber hinaus zeigt sich bei Problemen mit der Talkerfunktion aber auch häufig, daß die Eigenarten des Talkergerätes nicht genügend beachtet wurden. Man darf nicht vergessen, daß bei einem Meßgerät die Talkerfunktion auch von den meßtechnischen Funktionen des Gerätes abhängt. Das heißt, bei vielen Geräten müssen auch bestimmte meßtechnische Voraussetzungen erfüllt sein, bis eine Datenausgabe möglich ist. So müssen häufig Meßzeiten abgewartet, zulässige Meßgrenzen eingehalten oder zunächst Triggervorgänge aktiviert werden, um den Meßvorgang auszulösen.

Versucht man unzulässige Datenausgaben von dem Gerät zu erreichen, so kann sich dieses zunächst mit SRQ melden, um eine Fehlermeldung zu geben. Bei vielen Geräten ist dann die Talkerfunktion erst wieder möglich, wenn die SRQ-Meldung durch eine entsprechende Abfrage beantwortet wurde (siehe auch 6.5). Im allgemeinen läßt sich aber an der Frontplatte des jeweiligen Talkergerätes erkennen, ob die Talkerfunktion möglich ist oder nicht, weil die Meßwertanzeige im Gerät funktionsmäßig mit der Talkerfunktion häufig fast identisch ist.

6.5 Service Request

Die Funktion SRQ (*Service Request*) ist nicht in allen Meßgeräten enthalten, da sie zu den Optionen des IEC-Busses gehört. Sie gibt jedem Meßgerät die Möglichkeit, den Steuerrechner zu unterbrechen. SRQ wird im allgemeinen verwendet, wenn Meßgeräte dringende Meldungen an den Steuerrechner machen müssen, auch wenn dieser innerhalb seines normalen Programmes mit anderen Dingen beschäftigt ist. Dies gilt insbesondere für Fehlfunktionen von Geräten oder für das Erreichen von irgendwelchen Zuständen, bei denen der Rechner sein Programm ändern muß. Grundsätzlich können alle diese Aufgaben auch ohne SRQ gelöst werden, indem der Rechner zyklisch die in Frage kommenden Geräte abfragt. Allerdings bedeutet dies Zeitverlust und Komplizierung des Programms.

Man sollte deshalb ein IEC-Bus-System zunächst ohne SRQ einwandfrei zum Funktionieren bringen. Bei vielen Geräten besteht die Möglichkeit, die Funktion SRQ abzuschalten. Tut man dieses nicht, so muß dieses im Programm berücksichtigt werden. Andernfalls ergeben sich bei jedem SRQ eines Gerätes Störungen im Programmablauf, sofern der Steuerrechner die Fähigkeit besitzt, SRQ zu erkennen. In diesem Falle löst nämlich jede Aktivierung der SRQ-Leitung sofort beim Steuerrechner eine Unterbrechung des laufenden Programms hervor. Dann muß der Rechner aber wissen, was er beim Empfang einer SRQ-Meldung zu tun hat. Dazu besitzen entsprechende Rechner einen eigenen Programmbefehl, der z. B. lautet:

ON SRQ THEN ...

das heißt, im Falle SRQ soll der Rechner zur Zeile ... springen. Solange diese Definition im Programm nicht getroffen ist, gibt der Rechner eine Störungsmeldung, die besagt, daß er SRQ empfangen hat, aber nicht weiß, was in diesem Falle zu tun ist. Mit anderen Worten: das Vorhandensein von Geräten mit SRQ muß grundsätzlich im Programm berücksichtigt werden. Viele Meßgeräte erwarten darüber hinaus eine Reaktion auf ihre SRQ-Meldung. Vom Rechner muß die Serien- oder Parallelabfrage ausgelöst werden, um das Gerät zu identifizieren, das die SRQ-Meldung gegeben hat. Andernfalls wird das Gerät im SRQ-Zustand bleiben und seine übrigen Funktionen blockieren.

6.6 Wartezeiten innerhalb des Systems

Theoretisch erlaubt der IEC-Bus eine Datenübertragungsrate bis zu 1 MByte/s. Tatsächlich können aber bedingt durch meßtechnische oder geräteinterne Vorgänge innerhalb der Datenübertragung erhebliche Wartezeiten auftreten. Wenn z. B. ein Meßgerät zur Lieferung eines Meßergebnisses aufgefordert wird, so kann es sein, daß das Ergebnis aus meßtechnischen Gründen erst nach einer langen Wartezeit verfügbar ist.

In diesem Falle gibt es mehrere Möglichkeiten, um diese Wartezeit zu berücksichtigen:

6.6.1 Zeitkontrolle über Handshake

Das Meßgerät wird als Talker adressiert, und liefert das Ergebnis erst nach der Wartezeit. Dieses sehr einfache Verfahren hat aber den großen Nachteil, daß es für die Dauer der Wartezeit den Handshakezyklus und damit das ganze System blockiert, so daß die Wartezeit nicht für andere Aufgaben verwendet werden kann. Außerdem entstehen Probleme, wenn aus irgendwelchen Gründen die Durchführung der Messung nicht möglich ist, so daß die Meßzeit unendlich lang wird. In diesem Falle würde das gesamte Bus-System blockiert bleiben. Helfen kann hier nur noch, soweit vorhanden, die „Time out"-Funktion des Rechners (siehe 6.7).

6.6.2 Zeitkontrolle durch zyklische Abfragen

Das Steuergerät fragt das Meßgerät in bestimmten Zeitabständen über eine spezielle Talkerfunktion immer wieder, ob der Meßvorgang beendet ist, um dann das Ergebnis abzurufen. Hierbei wird eine Blockierung des Bus-Systems vermieden. Das Verfahren erfordert wegen der häufigen Abfragen jedoch erheblichen zusätzlichen Datenverkehr und kann außerdem den Zeitpunkt der Beendigung des Meßvorganges nur ungenau erfassen.

6.6.3 Zeitkontrolle über Service Request

Das Steuergerät aktiviert die Meßfunktion des Gerätes und fragt das Ergebnis erst ab, wenn das Meßgerät durch SRQ die Beendigung der Messung gemeldet hat. In der Zwischenzeit sind Bus- und Steuergerät völlig frei für andere Aufgaben.

Es ist offensichtlich, daß, soweit technisch möglich, die dritte Möglichkeit vorzuziehen ist.

6.7 Zeitüberwachung des Steuergerätes

Aus dem bisher Gesagten ergibt sich, daß der IEC-Bus durch verschiedene Ursachen blockiert werden kann. In diesem Falle würde das gesamte System funktionsunfähig werden und das Steuergerät wäre nicht einmal in der Lage, eine Fehlermeldung abzugeben. Aus diesem Grunde haben einige Rechner eine Wartezeitbegrenzung eingebaut. Das heißt, es wird beim Handshakezyklus nicht beliebig lange auf die Antwort der Geräte gewartet. Vielmehr wird nach Überschreiten einer Maximalzeit zum nächsten Zyklus übergegangen. Diese Steuergeräteeigenschaft ist sehr nützlich, wenn die Wartezeit variiert werden kann und wenn das Überschreiten der Wartezeit im Programmablauf erkannt werden kann. Es gibt aber auch Rechner (z. B. Commodore PET), die nur eine feste kurze Wartezeit haben, und deren Überschreiten nicht ohne weiteres erkennbar ist. In diesem Falle ist große Vorsicht notwendig, da bei kurzen Wartezeiten unter Umständen Daten verloren gehen können.

a.	b.
Listeneradresse setzen	Listeneradresse setzen
Daten übertragen	Daten übertragen
Listeneradresse löschen	Talkeradresse setzen
Talkeradresse setzen	Daten übertragen
Daten übertragen	
Talkeradresse löschen	

Bild 10. Adressierungsreihenfolge a) ohne, b) mit automatischer Entadressierung

6.8 *Hilfsmittel zur Kontrolle der IEC-Bus-Funktionen*

Bei Austesten eines IEC-Bus-Systems ist der dabei eingesetzte Steuerrechner im allgemeinen ein wertvolles Hilfsmittel, um Informationen über Zustand und Funktion des Systems zu bekommen.

6.8.1 Bestimmung von unbekannten Geräteadressen

Wird ein nicht vorhandenes IEC-Bus-Gerät durch einen Programmbefehl angesprochen, oder wird für ein vorhandenes Gerät eine andere als die eingestellte Adresse aufgerufen, so gibt der Rechner eine Fehlermeldung in der entsprechenden Programmzeile. Durch Angabe dieser Zeilennummer ermöglicht der Rechner dabei einfachste Lokalisierung der Programmstelle, an der die falsche Adresse aufgerufen wurde.

Auf ähnliche Weise ist es einfach möglich, bei einem Gerät mit unbekannter Adresse festzustellen, auf welche Adresse dieses Gerät eingestellt ist:

Man verbindet das zu untersuchende Meßgerät mit dem Steuergerät, wobei alle anderen IEC-Bus-Geräte abgetrennt sind. Nun sendet man zu diesem Gerät Listenerbefehle, wobei die Adresse bei 1 beginnend solange erhöht bis keine Fehlermeldung mehr auftritt. Man kann dieses auch in Form eines einfachen Programmes vornehmen:

```
10  X=X+1
20  IF X > 30 THEN 60
30  PRINT @ X: "XYZ"
40  PRINT X
50  END
60  PRINT "Gerät nicht ansprechbar"
70  END
```

Vor dem ersten Starten des Programms muß der Wert X auf 0 gesetzt werden. Sodann startet man das Programm mit dem Befehl „RUN". Im Falle einer fehlerhaften Adresse wird das Programm mit der Fehlermeldung „Adresse nicht gefunden" in Zeile 30 stehen bleiben. Durch erneutes Starten des Programmes wird der gleiche Vorgang für die nächste Adresse wiederholt. Dies geschieht bis zum Erreichen der richtigen Adresse, die dann auf dem Bildschirm ausgedruckt wird. Je nach Eigenschaften des Rechners kann man das Programm natürlich auch vollautomatisch gestalten.

6.8.2 Das Steuergerät als Monitor für Datenausgabe

Eine wesentliche Erleichterung für die Analyse von IEC-Bus-Funktionen ist die Tatsache, daß normalerweise zum Datenverkehr der ISO-7-Bit-Code verwendet wird. Das bedeutet, daß fast allen übertragenen Zeichen direkt lesbare Buchstaben und Zahlensymbole entsprechen. Diese Tatsache kann man sich auf einfache Weise bei der Analyse des IEC-Bus-Datenverkehrs zunutze machen:

Jedem zu analysierenden IEC-Bus-Ausgabebefehl des Steuergerätes, der eine variable Größe ausgibt, wird ein zusätzlicher PRINT-Befehl mit der gleichen Ausgabevariablen und dem gleichen Ausgabeformat zugeordnet. Zum Beispiel:

```
100  PRINT @ 17:USING 730:A –    IEC-Bus-Befehl
110  PRINT USING 730:A –         Monitorbefehl
```

Auf diese Weise wird auf dem Bildschirm des Steuergerätes eine vollkommen identische Kopie der vom Steuergerät ausgegebenen Daten protokolliert. Dies ist insbesondere bei der Verwendung von formatierten Ausgabebefehlen von entscheidender Bedeutung, da es häufig schwierig ist, die Datenausgabeformate derartiger Befehle, insbesondere im Hinblick auf Anzahl der Space, Positionierung der Vorzeichen usw. zu kennen.

Sollen während eines laufenden Programmes mehrere Ausgabebefehle gleichzeitig am Bildschirm mitgeschrieben werden, empfiehlt es sich, den Monitorbefehl durch eine zusätzliche Kennung zu markieren, damit eindeutig ist, welche der auf dem Bildschirm ausgegebenen Daten zu welchem IEC-Bus-Befehl gehört: z. B.:

```
110  PRINT: USING 730: A; "Ausgabe A"
```

In diesem Falle wird in der gleichen Bildschirmzeile, in der die auf dem IEC-Bus ausgegebene Größe A dargestellt wird, zusätzlich die Information „Ausgabe A" ausgedruckt.

6.8.3 Das Steuergerät als Monitor für Datenempfang (siehe auch 6.4.2).

Für die Überwachung der durch das Steuergerät empfangenen IEC-Bus-Daten empfiehlt sich ebenfalls die Kombination des jeweiligen IEC-Bus Input-Befehles mit einem Print-Befehl, der die gleiche Variable auf dem Bildschirm darstellt. Beispiel:

```
INPUT @ 17: B$
PRINT B$; "Einlesen B"
```

In diesem Falle wird auf dem Bildschirm die an der entsprechenden Programmstelle eingelesene Größe B$ mit der Bemerkung dargestellt, daß es sich um die „Einlesegröße B" handelt.

6.8.4 Statische Funktionskontrolle des IEC-Bus

Es gibt inzwischen eine Vielzahl von mehr oder weniger aufwendigen IEC-Bus-Meßgeräten auf dem Markt. In vielen Fällen kommt man jedoch mit wesentlich einfacheren Hilfsmitteln aus: Da der IEC-Bus ein asynchrones System ist, läßt er sich durch entsprechende Verzögerung der Handshake-Leitungen beliebig verlangsamen.

Dazu simuliert man den Handshakezyklus, indem man der NRFD-Leitung und der NDAC-Leitung je einen Schalter nach 0 V parallel schaltet (siehe *Bild 11*). Sind beide Schalter geschlossen, wird die Funktion des Systems zunächst gesperrt. Dann wird der NRFD-Schalter geöffnet, es erscheint das erste Datenwort auf dem Bus und DAV wird aktiv. Nun wird der NRFD-Schalter wieder geschlossen und der NDAC-Schalter geöffnet. Dies bedeutet, daß der Empfang der Daten beendet ist. Die Daten verschwinden von den Bus-Leitungen und das System wartet auf das nächste NRFD-Signal. Das heißt, mit zwei einfachen Schaltern läßt sich der vollständige Handshakezyklus simulieren, so daß der Datenfluß des IEC-Bus beliebig verzögert werden kann. Mit diesem Hilfsmittel ist es möglich, die einzelnen Datenworte statisch z. B. mit Hilfe eines Logiktesters zu betrachten.

Ergibt sich dabei die Situation, daß bei langsamen handgesteuerten Handshake fehlerfreie Listener- und Talkerfunktionen festgestellt wird, während bei Betrieb mit normaler Geschwindigkeit keine einwandfreie Datenübertragung vorliegt, so ist dies ein eindeutiger Hinweis darauf, daß alle IEC-Bus-Funktionen, Adressen und Programme grundsätzlich richtig sind, daß aber dynamische Probleme im Bereich der Datenübertragung auftreten, die durch die Verkabelung, Anzahl der Geräte oder aber auch fehlerhafte Geräte hervorgerufen werden können.

7 Optimale Programmgestaltung

Die Erstellung von Steuergeräteprogrammen für IEC-Bus-Systeme stellt insbesondere, bei Verwendung von modernen, intelligenten Meßgeräten grundsätzlich keine große Schwierigkeit dar. Einige Punkte können jedoch, wenn sie rechtzeitig beachtet werden, die Effektivität der Programme wesentlich verbessern.

7.1 *Variable Adressen*

Soweit von den Eigenschaften des Steuergerätes her möglich, sollte man Variable als Adressen verwenden. Zum Beispiel:
Nicht: PRINT @ 17: "XYZ"
Sondern: PRINT @ X: "XYZ" (X=17)
In diesem Falle kann auch innerhalb eines langen Programmes durch Ändern des Inhalts einer einzigen Variablen eine Geräteadresse beliebig umgeschaltet werden.

7.2 *Zeitoptimierung*

Bei BASIC-Programmen wird für die Ausführung einer jeden Programmzeile eine bestimmte Zeit benötigt. So erreicht man eine Zeitoptimierung von Programmen dadurch, daß zeitintensive Rechenvorgänge parallel zu zeitintensiven Meßvorgängen laufen.

Dies erreicht man, indem man *erst den Meßvorgang* im Meßgerät aktiviert, um erst *dann parallel die Rechneroperationen* im Steuergerät durchzuführen.

Bei üblichen IEC-Bus-Steuergeräten werden Daten innerhalb eines Strings mit wesentlich höherer Datenfolge ausgegeben werden, als Daten, die durch hintereinander folgende Zeilen ausgegeben werden.
Beispiel:
Durch den Befehl PRINT @ 17: "ABC"
wird eine wesentlich kürzere Ausgabezeit für die Zeichenfolge ABC erreicht als durch die Befehlsfolge:

PRINT @ 17: "A";
PRINT @ 17: "B";
PRINT @ 17: "C".

7.3 *Ausgabeformate*

Während die Ausgabe von festen Codezeichen unproblematisch ist, muß die Ausgabe von Variablen mit besonderer Sorgfalt behandelt werden. Sofern vorhanden, empfiehlt sich die Verwendung eines Ausgabebefehls mit Formatierungsanweisung. Dabei sind die für den jeweiligen Rechner gültigen Formatierungsanweisungen genau zu beachten. Insbesondere bei Ausgabe von Zahlen werden häufig zusätzliche

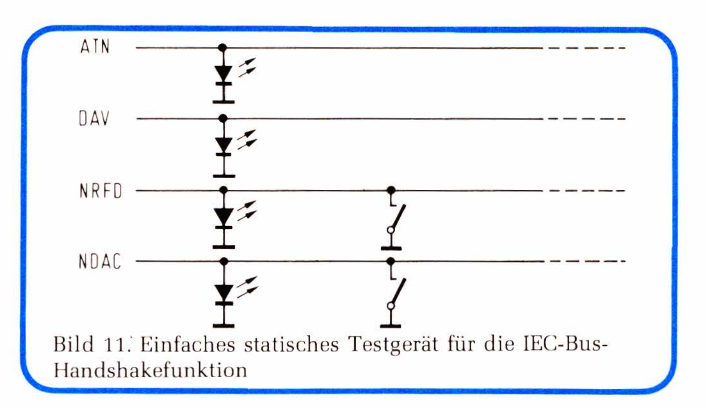


Bild 11: Einfaches statisches Testgerät für die IEC-Bus-Handshakefunktion

Zeichen wie z. B. Spacezeichen mit ausgesandt, die u. U. zu zunächst nicht erkennbaren Störungen der Gerätefunktionen führen können. Wichtig ist ferner, ob der Steuerrechner die Ausgabe von Daten an den IEC-Bus mit einem Schlußzeichen (meistens CR oder LF) abschließt oder nicht. Dieses Ausgabeformat muß mit den Erfordernissen der einzelnen Meßgeräte übereinstimmen, die teilweise Befehle sofort nach Empfang ausführen, teilweise aber auch erst, wenn ein Schlußzeichen die Befehlsfolge abgeschlossen hat. Die meisten Geräte werden auch dann nicht in ihrer Funktion gestört, wenn sie kein Schlußzeichen benötigen und dieses aber trotzdem gesendet wird.

Besondere Vorsicht ist bei Meßgeräten geringerer Intelligenz geboten, die zur Einstellung eine feste Anzahl von Ziffern verlangen, das heißt, die nicht in der Lage sind, auf führende Nullen zu verzichten. Beispiel:

Ein NF-Generator verlangt die Eingabe der Frequenz in Hz mit fünf Stellen und nachfolgendem F: Da das Gerät selber nicht in der Lage ist, führende Nullen zu ergänzen, so müssen diese unbedingt vom Steuergerät erzeugt werden.
Frequenzeinstellung 90 kHz
PRINT @ 17: "90000F"
Frequenzeinstellung 900 Hz
PRINT @ 17: "00900F"
dagegen nicht zulässig PRINT @ 17: "900F"
Programmtechnisch realisiert man dieses am einfachsten, indem man im Programm zunächst den Befehl:
PRINT @ 17: "000000"
einfügt. Danach darf man dann den Befehl
PRINT @ 17: "900F"
verwenden. Die meisten Geräte, die keine führenden Nullen ergänzen, arbeiten nämlich mit einem schieberegisterähnlichen Eingangsbuffer, so daß zulässig ist, soviele führende Nullen einzugeben, wie sie im ungünstigsten Falle benötigt werden.

Natürlich sind in dieser Zusammenstellung nicht alle Schwierigkeiten erfaßt, die einem Benutzer bei der Anwendung des IEC-Bus begegnen können. Sie soll aber dazu beitragen, daß der Anwender nicht ohne jedes Praxisgefühl seine ersten Schritte in die IEC-Bus-Technik macht und gleichzeitig auch demonstrieren, daß die möglicherweise auftretenden Probleme meist einfacher Natur sind, und daß der IEC-Bus ein leicht zu handhabendes Mittel ist, um Meßgeräte zu automatischen Testsystemen zu integrieren.

Dr. Manfred Richter

Das Zustandsdiagramm und seine Anwendung beim IEC-Bus

Durch die Verbreitung des IEC-Bus ist die Methode der Schaltungsentwicklung mit Zustandsdiagrammen wieder etwas mehr in den Blickpunkt der Entwickler geraten. Im Prinzip ist diese Methode altbekannt – die ersten Veröffentlichungen erschienen schon 1954 [1]; die ELEKTRONIK befaßte sich 1973 damit [2]. Auch die Funktion des Busses wurde unseren Lesern sorgfältig erklärt [4]. Schaltungsbeispiele zeigten den Anschluß von Wortgeneratoren [5] und von Prozeßrechnern [6] an den IEC-Bus. Bisher fehlte jedoch eine allgemein gehaltene Anleitung, wie man aus einem Zustandsdiagramm eine Schaltung entwickeln kann (implementieren); dies wird hier Schritt für Schritt gezeigt.

1 Das Zustandsdiagramm

1.1 Grundbegriffe

Unter einem Zustand versteht man eine „Verarbeitungskapazität" eines Systems. Etwas konkreter: ein System kann in einem bestimmten Zustand eine Nachricht empfangen, kann dadurch in einen anderen Zustand übergehen, und in jedem Zustand können bestimmte Nachrichten verschickt werden. Grafisch ist dies in *Bild 1* dargestellt. In Zustand A wird die Nachricht X empfangen, dadurch wird Zustand B aktiv, in B wird die Nachricht Y versendet.

Ein einfaches Beispiel aus dem Leben: Unsere Augen haben zwei mögliche Zustände: offen und zu. Durch einen Nervenimpuls X kann man das geschlossene Auge A in den offenen Zustand B überführen, hier wird dem Hirn das Ausgangssignal Y übermittelt – man kann sehen.

Hier stößt man bereits auf neue Begriffe: X ist ein *Eingangssignal*, Y ein *Ausgangssignal*, beide Nachrichten werden auch *Variable* genannt. Jede Variable hat wieder zwei mögliche Zustände: *wahr* oder *falsch*. Man kann beispielsweise sagen: im Zustand B ist die Ausgangsvariable Y wahr, in A aber falsch (statt „falsch" sagt man oft auch „unwahr").

Es gelten folgende Regeln:

- *Jede Variable kann aus einem Boole'schen Ausdruck von mehreren Variablen bestehen.*
- *Jeder Zustand kann aus mehreren Unterzuständen bestehen und daher in diese zerlegt werden, auch kann man mehrere Zustände zu einem zusammenfassen.*

1.2 Anwendung auf elektronische Schaltungen

Bild 2 zeigt ein ganz allgemeines System, dies kann eine elektronische Schaltung sein. Sind die Ausgangssignale $A_0...A_j$ nur Funktionen der Eingangssignale $E_0...E_i$, spricht man von einer *kombinatorischen* Schaltung. Beispiel: ein UND-Glied. Die Ausgangssignale sind durch die Eingangssignale vollständig definiert; die Laufzeit sei vernachlässigt. Bei einer *sequentiellen* Schaltung sind die Ausgangssignale nicht nur Funktionen der Eingangssignale, sondern auch der vorhergegangenen Ausgangssignale; sie sind damit auch von der *Zeit* abhängig. Beispiel: ein Flipflop. Die Ausgangssignale sind nur definiert, wenn neben den neuen Zuständen der Eingänge auch die vorhergehenden Zustände der Ausgänge bekannt sind.

In *Bild 3* ist ein einfaches Beispiel für das Zustandsdiagramm einer kombinatorischen Schaltung dargestellt. Dies ist immer daran zu erkennen, daß die Übergangsvariablen von A nach B und zurück einander Inverse sind. Bei *Bild 4* handelt es sich um eine sequentielle Schaltung. Hier können die Variablen X und Y zugleich falsch sein. Das Diagramm eines RS-Flipflops kann so aussehen. Sind X und Y zugleich wahr, oszilliert die Schaltung.

1.3 Schaltungsentwicklung

Die Entwicklung einer Schaltung kann man in folgende Schritte zerlegen:

1) Erstellung eines Flußdiagrammes
2) Erstellung des Zustandsdiagrammes
3) Überführung in ein Karnaugh-Diagramm (KV-Tafel)

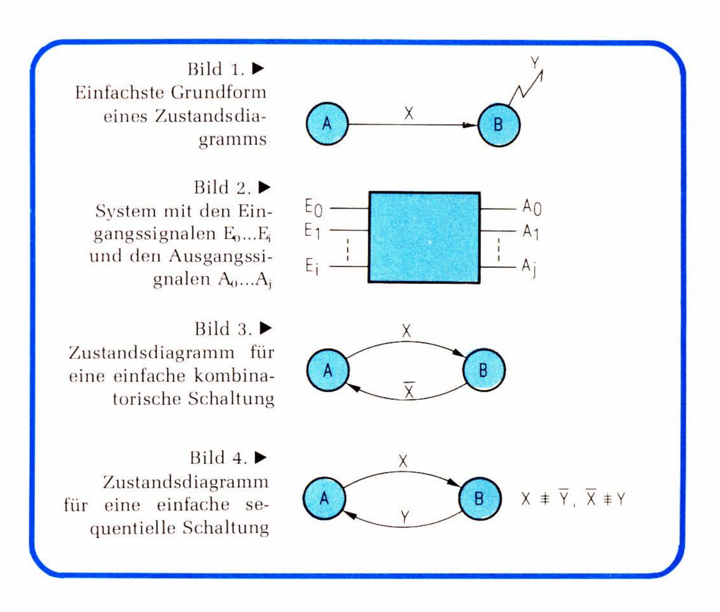


Bild 1. ▶ Einfachste Grundform eines Zustandsdiagramms

Bild 2. ▶ System mit den Eingangssignalen $E_0...E_i$ und den Ausgangssignalen $A_0...A_j$

Bild 3. ▶ Zustandsdiagramm für eine einfache kombinatorische Schaltung

Bild 4. ▶ Zustandsdiagramm für eine einfache sequentielle Schaltung

4) Aufstellung der Gleichungen für die Flipflop-Eingänge und die Schaltungs-Ausgänge
5) Zeichnen der Schaltung.

Das *Flußdiagramm* ist eine übersichtliche und leicht verständliche Methode zur Funktionsbeschreibung der Schaltung. Mit den bekannten Symbolen zur Software-Erstellung kann man das gewünschte Verhalten der Schaltung beschreiben. Bei einfachen Schaltungen, aber auch wenn man schon etwas mehr Übung hat, wird man diesen Schritt jedoch meist übergehen können. Das Zustandsdiagramm wird dann direkt aus der Definition der Schaltung heraus entworfen. Ein Flußdiagramm aber muß man erst in das Zustandsdiagramm übersetzen. Oft wird man dabei intuitiv vorgehen müssen. Im allgemeinen aber kann man sagen, daß eine Verzweigung meist einem Zustand entspricht.

Das *Zustandsdiagramm* wird nun durch Zusammenfassen so in Gruppen zerlegt, daß innerhalb einer Gruppe nur mehr sequentielle Zustände vorkommen. Man spricht dann vom „inneren Zustand" der Schaltung, im Gegensatz zum „totalen Zustand", der auch die kombinatorischen Zustände enthält.

Beim Erstellen des Zustandsdiagrammes ist besonders darauf zu achten, daß man nicht von einem Zustand zur selben Zeit zu mehreren anderen Zuständen gelangen kann. Dies könnte vorkommen, wenn mehrere Eingangsvariable zugleich wahr werden können.

Die Zustände werden nun in eine *KV-Tafel* eingetragen. Dadurch wird jedem Zustand ein Code zugeordnet. Die Übergänge werden durch Pfeile eingezeichnet. Dabei bedeutet ein Übergang in ein benachbartes Feld, daß genau ein Flipflop schaltet, bei einem schrägen Übergang schalten genau zwei Flipflops, bei einem „Rösselsprung" schalten drei Flipflops, usw. Die Zuordnung der Felder der KV-Tafel zu den Zuständen sollte so erfolgen, daß die Übergänge nach Möglichkeit ins benachbarte Feld gehen, dadurch wird der Schaltungsaufwand verringert.

Bild 5 zeigt ein Beispiel für die Zuordnung. Hier sollen JK-Flipflops verwendet werden. Aus der KV-Tafel können direkt die Bedingungsgleichungen für die J- und K-Eingänge abgelesen werden. In den Zuständen S 1 und S 0 ist Flipflop A eingeschaltet, in den Zuständen S 2 und S 3 ist dies Flipflop B. Um A einzuschalten, muß der J-Eingang von A (also JA) wahr werden. Dies gilt in Bild 5 für Übergänge von links nach rechts. Um A auszuschalten, muß K von A wahr werden, dies sind die Übergänge von rechts nach links. B wird eingeschaltet beim Übergang von oben nach unten und ausgeschaltet von unten nach oben. JA muß also wahr werden, wenn sich die Schaltung im Zustand S0 befindet und die Eingangsvariable U wahr wird, daher: $JA = S\,0 \wedge U$.

KA muß wahr werden, wenn S 1 aktiv ist, und W wahr wird, aber auch wenn S 3 aktiv ist und Y wahr wird. Dies ist also eine ODER-Verknüpfung:

$$KA = (S\,1 \wedge W) \vee (S\,3 \wedge Y)$$

Ebenso erhält man die Bedingungen für B:

$$JB = S\,1 \wedge X$$
$$KB = S\,2 \wedge Z$$

In den Gleichungen kommen die vier Zustände S 0...S 3 vor, die erst decodiert werden müssen:

$$S\,0 = \overline{A} \wedge \overline{B}$$
$$S\,1 = A \wedge \overline{B}$$
$$S\,2 = \overline{A} \wedge B$$
$$S\,3 = A \wedge B.$$

Man sieht, daß eine Schaltung im allgemeinen aus drei Funktionsblöcken besteht: Die Bedingungslogik, die Flipflops und die Zustandsdecodierung (*Bild 6*). Die Ausgangssignale der Schaltung sind entweder einzelne Zustände oder ODER-Verknüpfungen mehrerer Zustände.

Beim besprochenen Beispiel wurde davon ausgegangen, daß die Takteingänge aller Flipflops durch den selben Takt angesteuert werden; man spricht dann von *synchroner* Logik. Im Gegensatz dazu steht die *asynchrone* Logik, dabei schaltet meist ein Flipflop das nächste.

Eine Eigenschaft des JK-Flipflops kann man benützen, um die Schaltung noch zu vereinfachen: Nach dem Übergang von S 0 nach S 1 wird JA sofort falsch, da ja gilt: $JA = S\,0 \wedge U$, und S 0 wird ja inaktiv und daher falsch. Aber auch in S 1 darf JA noch wahr sein, der Zustand des Flipflops A wird dadurch nicht beeinflußt. Man erhält also eine neue Bedingungsgleichung: $JA = (S\,0 \wedge U) \vee (S\,1 \wedge U)$. Hier kann man U herausheben: $JA = U \wedge (S\,0 \vee S\,1)$.

Nun gilt aber: $S\,0 \vee S\,1 = \overline{B}$. Man kann daher für JA schreiben:

$$JA = \overline{B} \wedge U.$$

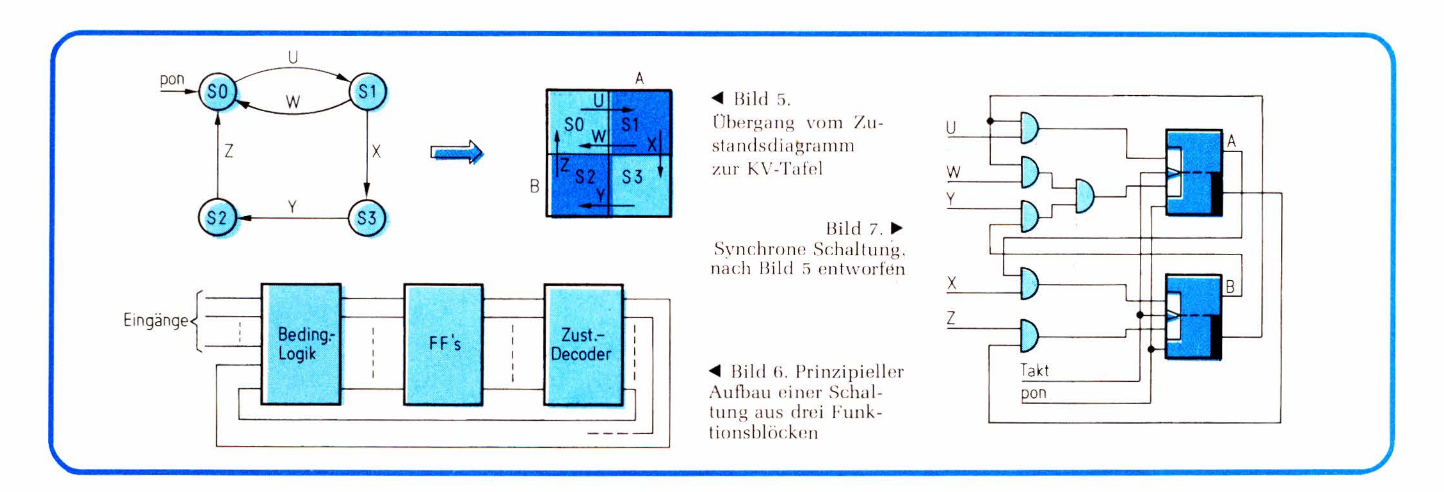
Ebenso erhält man für die anderen Flipflopeingänge:

$$KA = (\overline{B} \wedge W) \vee (B \wedge Y)$$
$$JB = A \wedge X$$
$$KB = \overline{A} \wedge Z.$$

Man sieht, daß auf diese Art der Zustandsdecodierer eingespart wurde. Ob dieser Weg der ökonomischere ist, muß von Fall zu Fall untersucht werden. Nun kann man die Schaltung zeichnen (*Bild 7*). Dabei werden die Nachrichten mit der Zuordnung

wahr ≙ 1
falsch ≙ 0

codiert. 0 und 1 sind die beiden Zustände eines Bits. Zur Codierung von Nachrichten werden oft Bitkombinationen



◄ Bild 5. Übergang vom Zustandsdiagramm zur KV-Tafel

Bild 7. ► Synchrone Schaltung, nach Bild 5 entworfen

◄ Bild 6. Prinzipieller Aufbau einer Schaltung aus drei Funktionsblöcken

verwendet. Ein Beispiel aus dem IEC-Bus ist die Nachricht UNL; sie ist wahr, wenn das Bitmuster 0111111 eintrifft, für alle anderen Bitkombinationen aber falsch.

Selbstverständlich wird man noch darauf achten, ob man alle Gleichungen nach den Regeln der Schaltalgebra vereinfachen kann.

1.4 Zusätzliche Hinweise zur Schaltungsentwicklung

Im folgenden sollen zunächst ein paar zusätzliche Regeln gegeben werden. Die meisten Schaltungen enthalten einen *Anfangszustand*, beim IEC-Bus Idle-state genannt. Beim Einschalten des Gerätes soll genau dieser Zustand aktiv werden. In Bild 5 ist dies der Zustand S 0, angedeutet durch den Pfeil mit der Variablen pon (power on). Die Schaltung in Bild 7 zeigt die Implementierung dieser Bedingung. An den asynchronen Rücksetzeingängen der Flipflops liegt das Eingangssignal pon, dieses wird nach dem Einschalten des Gerätes kurz wahr, um dann konstant falsch zu bleiben.

Ungebrauchte Zustände sollte man in den Anfangszustand oder einen benachbarten Zustand überführen. Das System könnte schließlich durch einen Störimpuls in so einen Zustand gelangen; hätte der dann keinen Ausgang, säße die Schaltung fest.

Der *„Overriding"-Eingang*: Wird dieses Signal wahr, geht die Schaltung in den Zustand über, der durch den Eingang bezeichnet wird, unabhängig davon, in welchem Zustand die Schaltung sich gerade befindet. Im Beispiel *Bild 8* ist dies der Eingang IFC. Alle anderen Übergänge können daher nur stattfinden, wenn der Overriding-Eingang falsch ist, dies wird daher in den Zustandsdiagrammen nicht explizit angegeben. Oft kann man dieses dominierende Eingangssignal ebenso wie pon an den Rücksetzeingang der Flipflops legen.

2 Zustandsdiagramme beim IEC-Bus

2.1 Listener

Bild 8 zeigt das etwas vereinfachte Zustandsdiagramm des Listeners. Zuerst sei untersucht, welche Übergänge kombinatorisch und welche sequentiell sind. MLA und UNL können nie zugleich wahr sein, sie können aber zugleich falsch sein, sind daher nicht einander Inverse. Der Übergang von LIDS nach LADS und zurück ist daher sequentiell. ATN und ATN sind einander Inverse, der Übergang LADS – LACS ist also kombinatorisch. Für die beiden Zustände LIDS und LADS braucht man ein Flipflop, es sei mit L bezeichnet. Zunächst zeichnet man die KV-Tafel; sie besteht nur aus zwei Feldern (*Bild 9*). Die beiden Zustände sind L 0 und L 1. Wohl gilt: LIDS = L 0, man kann aber nicht sagen: LADS = L 1; Flipflop L ist ja in den beiden Zuständen LADS und LACS eingeschaltet. Der Unterschied zwischen diesen bei-

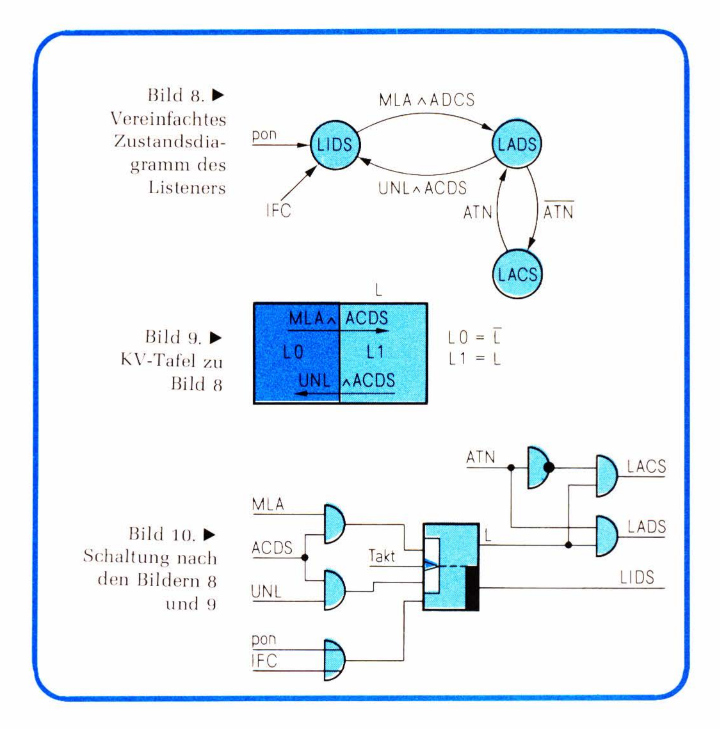


Bild 8. ▶ Vereinfachtes Zustandsdiagramm des Listeners

Bild 9. ▶ KV-Tafel zu Bild 8

Bild 10. ▶ Schaltung nach den Bildern 8 und 9

den wird durch ATN bestimmt. LADS kann nur aktiv sein, wenn ATN wahr ist, LACS kann nur aktiv sein, wenn ATN falsch ist. Durch kombinatorische Verknüpfung von L und ATN erhält man daher:

$$LADS = L \wedge ATN,$$
$$LACS = L \wedge \overline{ATN}.$$

Bild 10 zeigt die sich ergebende Listener-Schaltung.

2.2 Acceptor-Handshake (AH-Funktion)

Bild 11 zeigt wieder eine etwas vereinfachte Ausführung des Zustandsdiagramms. Das Signal tcs wurde weggelassen, es ist nur von Interesse, wenn man auch einen Controller implementiert hat. Man faßt zunächst die vier linken Zustände zu einem einzigen zusammen. Was dabei herauskommt, zeigt *Bild 12*. Man sieht, daß sich AIDS nur kombinatorisch von den anderen Zuständen unterscheidet, da ja ATN ∨ LA und $\overline{ATN} \wedge \overline{LA}$ einander Inverse sind. Zur Implementierung dieses Zustandes braucht man also keinen Flipflopzustand, so daß man für die ganze Schaltung sicher nicht mehr als zwei Flipflops braucht. Hinzu kommt noch, daß die Übergänge von und nach AIDS innerhalb der Zeit t 2 erfolgen müssen.

In einer sequentiellen Schaltung wäre dies nur mit einer viel zu hohen Taktfrequenz zu erreichen. Diese Einschränkung gilt übrigens auch für den Übergang von ACRS nach ANRS. Der Zusatz T 3 aber sagt, daß die Schaltung mindestens die Zeit T 3 im Zustand ACDS bleiben muß, bevor sie

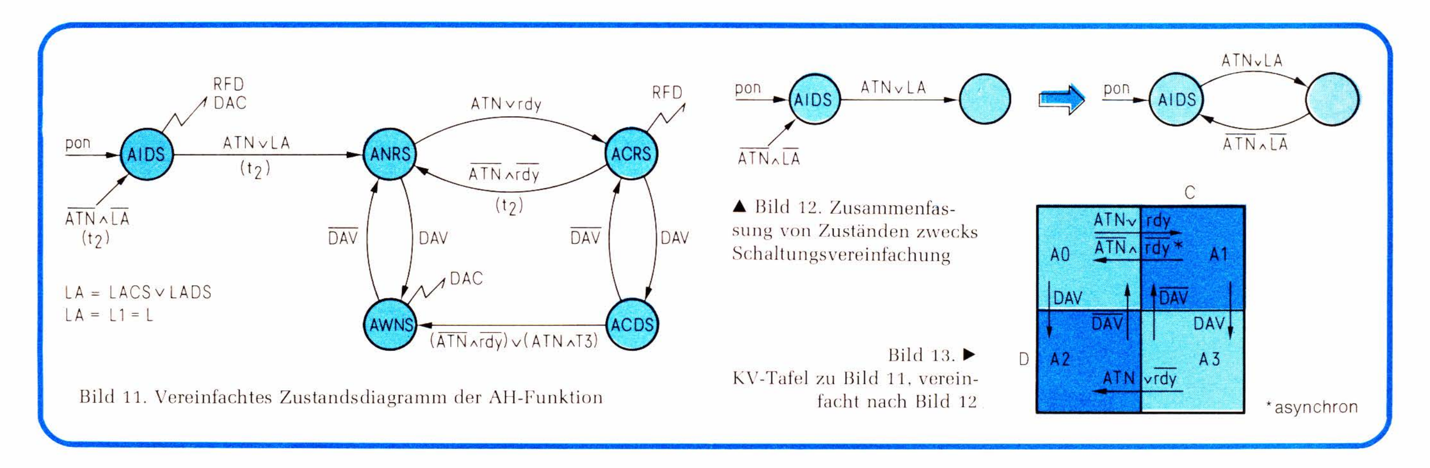


Bild 11. Vereinfachtes Zustandsdiagramm der AH-Funktion

▲ Bild 12. Zusammenfassung von Zuständen zwecks Schaltungsvereinfachung

Bild 13. ▶ KV-Tafel zu Bild 11, vereinfacht nach Bild 12

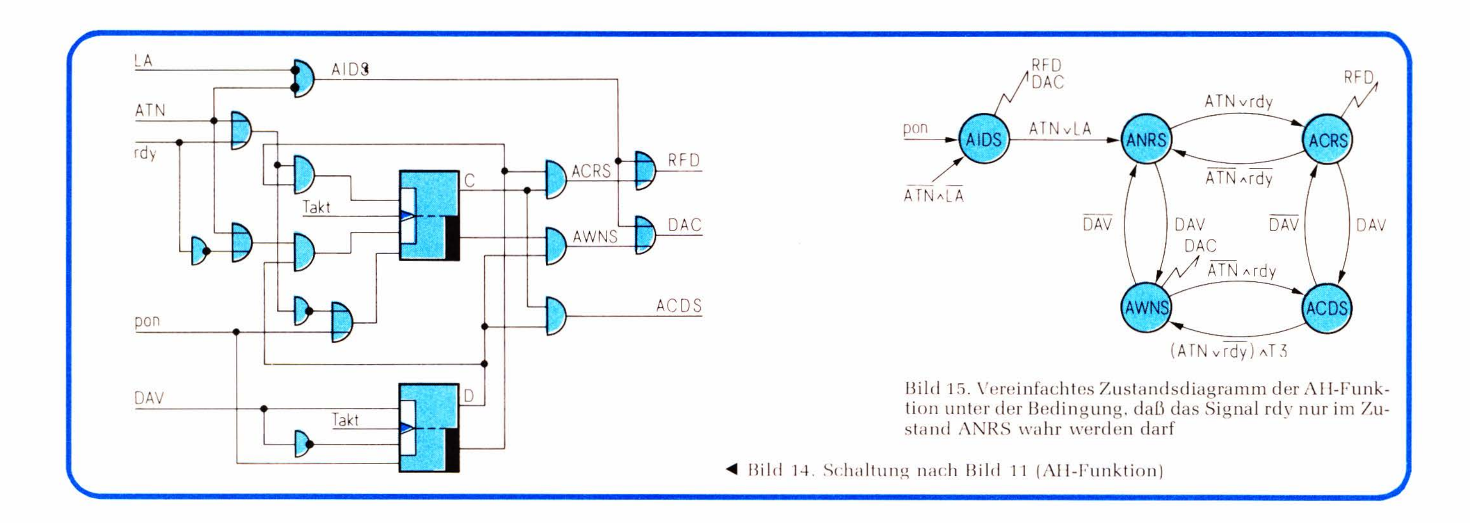


Bild 15. Vereinfachtes Zustandsdiagramm der AH-Funktion unter der Bedingung, daß das Signal rdy nur im Zustand ANRS wahr werden darf

◀ Bild 14. Schaltung nach Bild 11 (AH-Funktion)

mit ATN nach AWNS übergeht. Bei synchronen Schaltungen ist diese Bedingung automatisch erfüllt, die Schaltung bleibt ja dann genau eine Taktpriode in ACDS stehen. Aus dem Ausdruck beim Übergang ACDS – AWNS kann man also T 3 entfernen und man erhält:

$$(\overline{ATN} \wedge \overline{rdy}) \vee (ATN \wedge T\,3) = ATN \vee \overline{rdy}.$$

Nun kann man wieder die KV-Tafel zeichnen (*Bild 13*). Es ergeben sich folgende Bedingungsgleichungen:

$$JC = \overline{D} \wedge (ATN \vee rdy)$$
$$KC = D \wedge (ATN \vee \overline{rdy})$$
$$RC = (\overline{ATN} \wedge \overline{rdy}) \vee pon$$
$$JD = (\overline{C} \wedge DAV) \vee (C \wedge DAV) = DAV$$
$$KD = (\overline{C} \wedge \overline{DAV}) \vee (C \wedge \overline{DAV}) = \overline{DAV}$$
$$RD = pon.$$

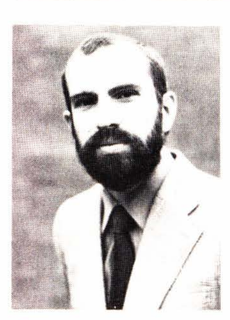
Die Übergänge von oben nach unten und zurück kann man natürlich auch kombinatorisch lösen, man kommt dann mit einem einzigen Flipflop aus. Wenn man hier aber trotzdem sequentiell arbeitet, dann darum, weil man auf diese Art ein digitales Filter für den Eingang DAV erhält. Die Anfälligkeit gegenüber Störungen auf der DAV-Leitung wird also verringert.

Nun zu den Ausgängen: RFD und DAC sind die zwei Busnachrichten, die von der Schaltung erzeugt werden. RFD ist in den beiden Zuständen AIDS und ACRS wahr, DAC in AIDS und AWNS. Diese Zustände müssen daher decodiert werden. In Bild 12 erkennt man, daß AIDS immer aktiv ist, wenn $\overline{ATN} \wedge \overline{LA}$ wahr ist. Daher gilt: AIDS = $\overline{ATN} \wedge \overline{LA}$. Die beiden anderen Zustände bekommt man auf die übliche Art:

$$ACRS = A\,1 = C \wedge \overline{D}$$
$$AWNS = A\,2 = \overline{C} \wedge D.$$

Für die Ausgänge selbst gilt:

$$RFD = AIDS \vee ACRS$$
$$DAC = AIDS \vee AWNS.$$



Dr. Manfred Richter ist in Wien geboren und studierte dort Physik und Mathematik. Nach einer kurzen Anstellung bei der Firma Viennatone in Wien ging er 1969 nach Holland zur Firma Philips. Dort entwickelte er digitale Meßgeräte und Meßsysteme. Es folgten zwei Jahre bei Philips in Hamburg; seit Oktober 1976 entwickelt er für die gleiche Firma in Wien Rechner- und Mikroprozessorsteuerungen für automatische Meßsysteme.
Seit 1970 war er mit der Entwicklung von Schnittstellen für Meßgeräte von Philips beauftragt und war von Beginn an Mitglied der WG 3 der IEC TC 66. Dort war er einer der Autoren des IEC-Bus-Dokumentes und war wesentlich an der Einführung der Zustandsdiagramme beteiligt. Hobbys: Klassische Musik, Skifahren, Bergsteigen, Fotografie. ELEKTRONIK-Leser seit 1965

Für andere Diagramme (z. B. den Listener) braucht man auch noch ACDS:

$$ACDS = C \wedge D.$$

Somit wurden alle Bedingungen für die Schaltung (*Bild 14*) abgeleitet. Mit dem Signal rdy (ready) teilt das Gerät der AH-Funktion mit, ob es die Nachrichten auf den DIO-Leitungen empfangen und verarbeitet hat. Ist rdy = wahr, so ist das Gerät bereit, neue Nachrichten zu empfangen. Durch rdy = falsch teilt das Gerät mit, daß die Nachricht empfangen und verarbeitet ist, aber es ist noch nicht bereit für eine neue Nachricht. Das ist es erst, wenn rdy wieder wahr wird. Unterwirft man rdy einer zusätzlichen Bedingung, so kann man die Schaltung (Bild 14) noch weiter vereinfachen. Diese Bedingung lautet: rdy darf nur im Zustand ANRS wahr werden. Dies läßt sich leicht erreichen, wenn die AH-Funktion dem Gerät die Gültigkeit von Nachrichten auf den DIO-Leitungen nicht durch ACDS, sondern durch DAV anzeigt.

Das neue Zustandsdiagramm zeigt *Bild 15*. Der Übergang von AWNS nach ACDS wurde hinzugefügt; fest steht aber, daß dieser nie aktiv werden kann, da ja rdy in AWNS nicht wahr werden kann. Der Übergang von ACDS nach AWNS wurde etwas geändert: die Verzögerung T 3 gilt nun immer, nicht nur wenn ATN wahr ist. Für die Funktion der Schaltung ist dies aber ohne Bedeutung. Jedenfalls sind nun alle parallelen Übergänge einander Inverse, so daß nun die ganze Schaltung rein kombinatorisch aufgebaut werden kann. Die Zeit T 3 erzeugt man durch Verzögerung des Eingangssignales DAV mit einem RC-Glied. Ein kombinatorischer Zustand ist genau dann aktiv, wenn alle Nachrichten, die in den betreffenden Zustand überführen, wahr sind. Mit DAV' gleich dem verzögerten DAV erhält man nun folgende Gleichungen:

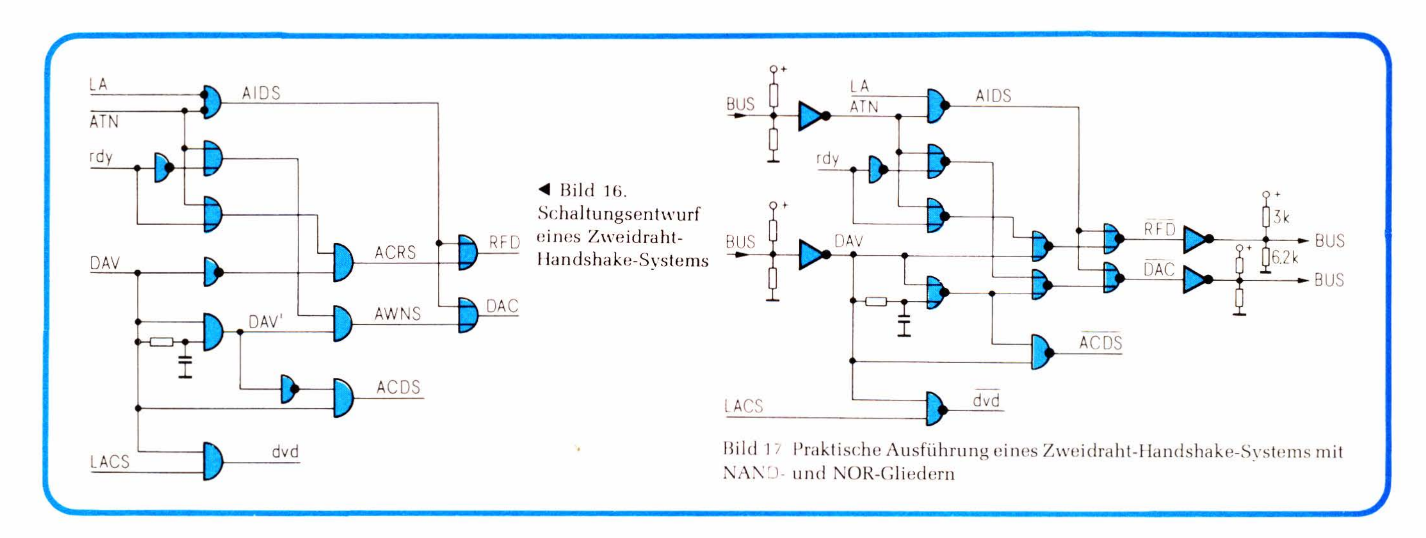
$$ACRS = (ATN \vee rdy) \wedge \overline{DAV'}$$
$$AWNS = (ATN \vee \overline{rdy}) \wedge DAV'$$
$$AIDS = \overline{ATN} \wedge \overline{LA}.$$

Für RFD und DAC gelten die selben Gleichungen wie oben. Für ACDS erhält man:

$$ACDS = DAV \wedge \overline{DAV'}.$$

2.3 *Zweidraht-Handshake-System*

Noch eine Erklärung zu *Bild 16*. Mit dvd (data valid for device) wurde das Signal bezeichnet, das den Gerätefunktionen die Gültigkeit von Nachrichten auf den DIO-Leitungen anzeigt. Die Signale rdy und dvd bilden zusammen ein Zweidraht-Handshake-System. Die praktische Ausführung in NAND's und NOR's zeigt *Bild 17*.



◀ Bild 16. Schaltungsentwurf eines Zweidraht-Handshake-Systems

Bild 17 Praktische Ausführung eines Zweidraht-Handshake-Systems mit NAND- und NOR-Gliedern

3 Schlußwort

Mit der hier beschriebenen Methode wurden die Schaltungen für die IEC-Bus-Schnittstellen der Philips-Geräte entwickelt, z. B. für das Digitalvoltmeter PM 2441, für die Meßstellenumschalter PM 2460 und für den Mosaikdrucker PM 2467. Ein Vergleich mit der Veröffentlichung [5], Bild 8, Seite 64, zeigt, wieweit man eine Schaltung durch Verwendung der Zustandsdiagramme und der hier besprochenen Methode bei der Entwicklung vereinfachen kann und wieviel integrierte Schaltungen sich einsparen lassen.

Literatur

[1] *Huffman, D. A.:* The synthesis of sequential switching circuits. Journal of the Franklin Institute, Bd. 257 (1954), H. 3, S. 161...190 und H. 4, S. 275...303.

[2] *Künzel, R.:* Das State-Diagramm. ELEKTRONIK 1973, H. 2, S. 47...52 und H. 3, S. 97...100.

[3] DIN-IEC 66.22: Elektrische Meßtechnik. Byteserielles, bitparalleles Schnittstellensystem für programmierbare Meßgeräte (Rosadruck). Beuth-Verlag GmbH, Berlin/Köln.

[4] *Klaus, J.:* Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.

[5] *Pannach, A.:* Interface-Entwicklung für den IEC-Bus. ELEKTRONIK 1975, H. 12, S. 61...64.

[6] *Richert, U.:* IEC-Bus-Interface für Prozeßrechner. ELEKTRONIK 1976, H. 12, S. 58...62.

Die Schnittstellenfunktionen ,,Bedienungsruf" und ,,Parallelabfrage" verleihen IEC-Bus-kompatiblen Geräten die Fähigkeit, sich aktiv bei der steuernden Einheit zu melden bzw. passiv an schnellen Statusabfragen teilzunehmen. Wie dies in Theorie und Praxis verwirklicht wird, zeigt der folgende Beitrag.

Dipl.-Inf. Heinrich Dietsch

Prinzip und Praxis der IEC-Bus-Funktionen „Bedienungsruf" und „Parallelabfrage"

1 Einführung und Überblick

Die mittlerweile erstaunlich weite Verbreitung der IEC-Bus-Schnittstelle ist in erster Linie denjenigen Herstellern von Meßgeräten, Tisch- und Minirechnern zu verdanken, die in relativ kurzer Zeit brauchbare Realisierungen der nicht immer leicht verständlichen Norm [1] fanden. Prinzipielle Möglichkeiten seien hier nur kurz erwähnt:

- Aufbau der Schnittstellenfunktionen mit SSI-MSI-Bausteinen bzw. FPLAs [2, 3, 4, 5],
- Software-Implementierung des Übertragungsprotokolls [6],
- Einsatz von LSI-Schnittstellenbausteinen [7].

Bild 1 zeigt ein allgemeines IEC-Bus-kompatibles Gerät: Die Gerätefunktionen senden in der Norm festgelegte „interne Nachrichten" (z. B. nba, rdy, rsv, ist) an die verwirklichte Teilmenge der Schnittstellenfunktionen. Die Norm ordnet den Zuständen der meisten Schnittstellenfunktionen feste Ausgangssignale zu; der Geräteentwickler kann jedoch weitere Signale als Rückmeldungen an die Gerätefunktionen hinzufügen. Die Gerätefunktionen kommunizieren im allgemeinen mit den 8 Bus-Datenleitungen und evtl. mit der Steuerleitung EOI.

Gegen den IEC-Bus werden oft die relativ niedrige Übertragungsrate (realistischer Wert: 200 kByte/s [4]) und die geringe Datenbusbreite ins Feld geführt. Diese Argumente verlieren jedoch an Bedeutung durch die Erkenntnis, daß die Leistung der derzeit verfügbaren IEC-Bus-Systeme nicht durch die Bus-Übertragungsrate, sondern durch die Fähigkeiten der angeschlossenen Geräte und Steuereinheiten begrenzt wird.

Gegenstand der vorliegenden Arbeit sind deshalb die Systemeigenschaften IEC-Bus-kompatibler Einheiten. Bei den meisten der auf dem Markt erhältlichen Geräte wurde auf die Realisierung der Schnittstellenfunktionen für „Bedienungsruf" und „Parallelabfrage" verzichtet, obwohl gerade damit Systemverhalten und Leistungsfähigkeit entscheidend zu beeinflussen wären. Der Grund ist wohl darin zu sehen, daß es für die Ansteuerung dieser Funktionen – im Gegensatz zu den Schnittstellenfunktionen für die Datenübertragung – kaum Hinweise und Anwendungsbeispiele gibt. In den Abschnitten 2 und 3 wird der Versuch gemacht, diese Schnittstellenfunktionen darzustellen und zu interpretieren.

Der darauffolgende Abschnitt beschreibt ein Gerätekonzept, das auf die Anforderungen der Prozeßrechentechnik ausgerichtet ist. Dabei ist zu betonen, daß keine „Norm in der Norm" formuliert werden soll; vielmehr sind Geräte, die die vorgeschlagenen Eigenschaften besitzen, auch in tischrechnergesteuerten Systemen einzusetzen. Abschließend wird der Ent-

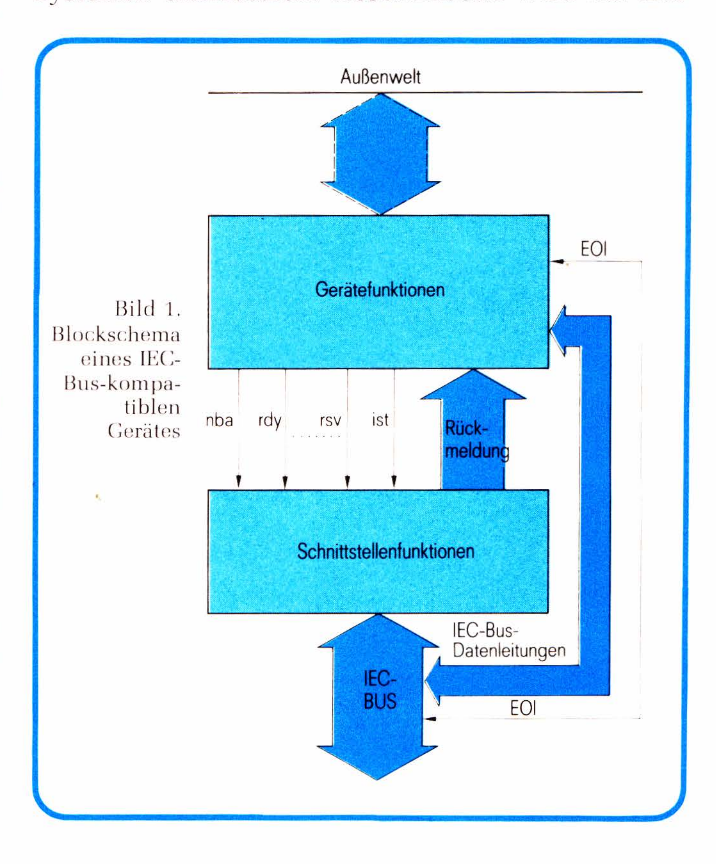
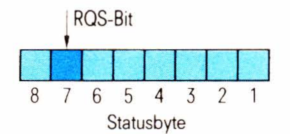


Bild 1. Blockschema eines IEC-Bus-kompatiblen Gerätes



RQS = 1: Gerät hat Bedienung angefordert
RQS = 0: Gerät hat keine Bedienung angefordert
Bits 1 ... 6,8 frei für gerätespezifische Meldungen

Bild 2. Das RQS-Bit im Statusbyte sagt aus, ob das Gerät Bedienung angefordert hat

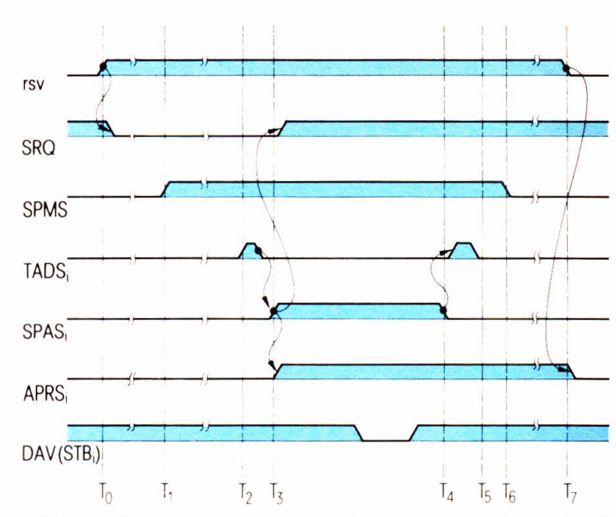


Bild 4. Ablauf: Bedienungsanforderung – Statusbyte-Abruf (Aktiv-Low-Pegel auf den Busleitungen SRQ und DAV)

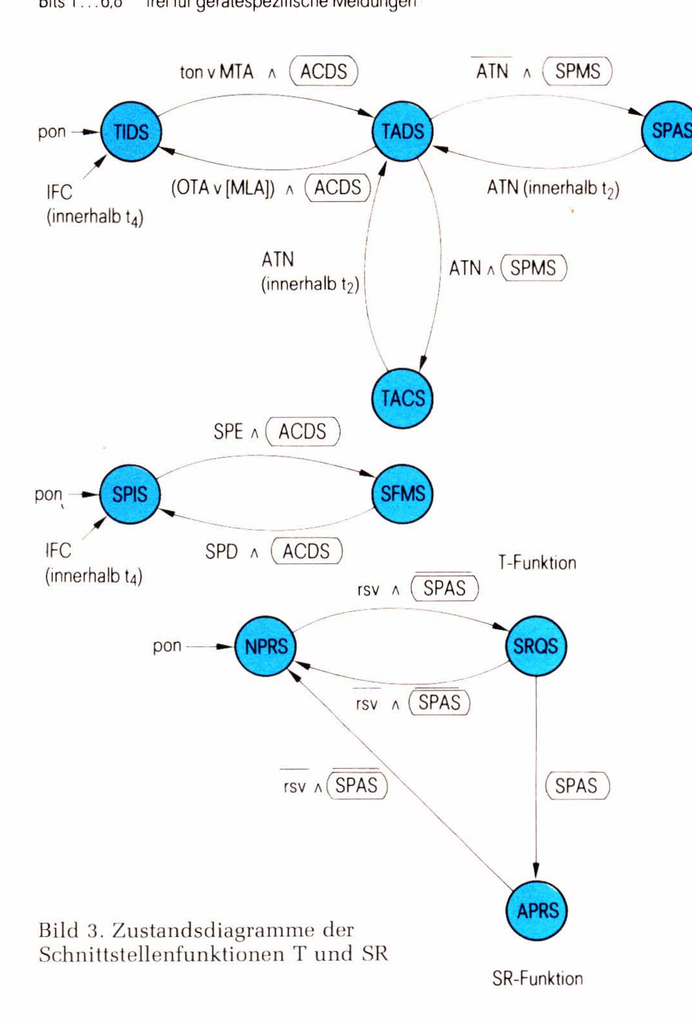


Bild 3. Zustandsdiagramme der Schnittstellenfunktionen T und SR

wurf einer universellen Schnittstellenschaltung vorgestellt, die die genannten Eigenschaften unterstützt.

2 Die SR-Schnittstellenfunktion

Ein Gerät, das mit einer SR-Funktion („Service Request" – „Bedienungsruf") ausgerüstet ist, kann über die SRQ-Busleitung asynchron eine Bedienungsanforderung an das Steuergerät absetzen. Weil alle N Geräte im Bus-System die nur einmal vorhandene SRQ-Leitung verwenden, repräsentiert diese zu jedem Zeitpunkt nur die logische ODER-Funktion aus den einzelnen Bedienungsanforderungen SRQ_i, $1 \leq i \leq N$:

$$SRQ = SRQ_1 \vee SRQ_2 \vee ... \vee SRQ_N \quad (1)$$

Jedes Gerät mit einer SR-Funktion muß auf Anfrage ein Statusbyte senden können. Das Statusbyte enthält neben der eigentlichen Nachricht das RQS-Bit, das aussagt, ob das betreffende Gerät Bedienung angefordert hat (*Bild 2*). Dem Steuergerät obliegt es dann, im Zuge einer seriellen Abfrage („Serial Poll") die Geräte herauszufinden, die eine Bedienungsanforderung abgesetzt haben.

Die SR-Funktion arbeitet eng mit der T-(„Talker", „Sprecher")-Funktion zusammen. *Bild 3* zeigt die Zustandsdiagramme der beiden Funktionen. *Bild 4* stellt in Form eines Impulsdiagramms den Ablauf von der Bedienungsanforderung eines Geräts i bis zum Abruf seines Statusbytes durch das Steuergerät dar: Die Gerätefunktionen des Geräts i wünschen zum Zeitpunkt T_0 Bedienung und melden dies mit der internen Nachricht rsv an ihre SR-Schnittstellenfunktion. Die SR-Funktion geht daraufhin vom Zustand NPRS („Negative Poll Response State") über in den Zustand SRQS („Service Request State"), in dem sie die Bedienungsanforderung über die SRQ-Busleitung sendet. In T_1 hat das Steuergerät die Anforderung zur Kenntnis genommen und startet mit dem Universalbefehl SPE („Serial Poll Enable") eine serielle Abfrage. SPE bringt die T-Funktionen aller Geräte in den Zustand SPMS („Serial Poll Mode State"). Im Intervall T_1T_2 werden sodann die Statusbytes mehrerer anderer Geräte ohne Erfolg auf das RQS-Bit hin untersucht, bis zum Zeitpunkt T_2 das Gerät i an der Reihe ist und zum Sprecher adressiert wird. Seine T-Funktion geht in den Zustand TADS („Talker Addressed State"), und schließlich, nachdem das Steuergerät das ATN-Signal zurückgenommen hat, wegen der Bedingung SPMS in den Zustand SPAS („Serial Poll Active State") (T_3). In T_3 muß den Gerätefunktionen über Rückmeldesignale mitgeteilt werden, daß ein Statusbyte gesendet werden soll. Die SR-Funktion geht (wegen SPAS) in den Zustand APRS („Affirmative Poll Response State"), dessen Ausgabesignal das RQS-Bit im Statusbyte darstellt (Bild 2). Es gilt:

$$RQS = APRS \wedge SPAS \quad (2)$$

Mit dem Verlassen von Zustand SRQS wird auch die Bedienungsanforderung auf der Busleitung SRQ zurückgenommen. Im Intervall T_3T_4 wird das Statusbyte im Handshake-Verfahren übertragen; nach Ende der Übertragung aktiviert das Steuergerät wieder die ATN-Busleitung (T_4), die T-Funktion geht zurück nach TADS, und zum Zeitpunkt T_5 verliert das Gerät i seine Sprechereigenschaft, z. B. durch Befehl UNT („Untalk") oder Erkennung von OTA („Other Talk Address"). Es wird angenommen, daß das Steuergerät nun die serielle Abfrage beendet: Der Universalbefehl

SPD („Serial Poll Disable") bewirkt in der T-Funktion den Übergang von SPMS nach SPIS („Serial Poll Idle State") (T_6). Erst nachdem die Gerätefunktionen die interne Nachricht rsv zurückgenommen haben (T_7), verläßt die SR-Funktion den Zustand APRS. Bis zu diesem Zeitpunkt bleibt bei weiteren Abfragen das RQS-Bit im Statusbyte gemäß (2) gesetzt.

Bild 5 zeigt Aufruf und Flußdiagramm einer allgemeinen seriellen Abfrageroutine SERPOLL für N Geräte, die an geeigneten Stellen in einem Hauptprogramm gestartet werden kann. Die Bedienzeiten für die einzelnen Geräte werden natürlich um so kürzer, je häufiger die Busleitung SRQ abgefragt wird. Zu beachten ist, daß die Abfrageroutine den „Adressierungszustand" des IEC-Bus-Systems (d. h. die aktuelle Verteilung der Hörer- und Sprecheradressen) verändert. Nach Beendigung der Routine müssen also, wenn dies der Ablauf erfordert, die vorher adressierten Geräte wieder neu adressiert werden.

SERPOLL macht zunächst das Steuergerät zum alleinigen Hörer und leitet dann mit dem Universalbefehl SPE die Abfrage ein. In einer Laufschleife wird jedes der N Geräte durch Sprecheradressierung angewählt und zum Senden seines Statusbytes aufgefordert. Ist das RQS-Bit im Statusbyte des i-ten Geräts gesetzt, legt SERPOLL die Nachricht in der i-ten Position der Tabelle STATAB ab. STATAB dient der Kommunikation mit dem rufenden Programm. Stellt sich nach einem Abfrageschritt heraus, daß sämtliche Bedienungsanforderungen erkannt worden sind, dann wird die Laufschleife verlassen, und der Universalbefehl SPD beendet die serielle Abfrage. Die Kontrolle geht wieder über an das rufende Programm, das auf den aktuellen durch STATAB repräsentierten Systemzustand geeignet reagieren kann. Bearbeitete Meldungen werden aus STATAB getilgt. Um speziellen Anforderungen zu genügen, könnte bereits in SERPOLL die Bearbeitung zeitkritischer Aufgaben vorgesehen sein (Beispiel: Sofortiges Abschalten eines Netzgeräts bei Überschreiten der Strombegrenzung).

Für die sinnvolle Verwendung von SERPOLL ist es wichtig, daß das RQS-Bit im Statusbyte nicht gesetzt bleibt, nachdem das Statusbyte nach der Bedienungsanforderung abgeholt wurde. Die interne Nachricht rsv muß also zurückgenommen werden, sobald der Zustand APRS vorliegt (T_3 in Bild 4). Nach erfolgter Übertragung des Statusbytes verläßt die T-Funktion den Zustand SPAS, und die SR-Funktion geht daraufhin in den Zustand NPRS, in dem RQS = „0" gilt.

SERPOLL verarbeitet also nur Statusbytes, die mit dem RQS-Bit gültig erklärt werden, d. h. „aktive" Zustandsmeldungen. Es ist einem Gerät durchaus gestattet, sein Statusbyte-Ausgaberegister ohne anschließende Bedienungsanforderung zu laden („passive" Zustandsmeldung), doch läßt sich diese Art Geräteinformation vor allem in komplexen Systemen nur schwer handhaben.

3 Die PP-Schnittstellenfunktion

Die PP-Schnittstellenfunktion („Parallel Poll", „Parallelabfrage"), deren Zustandsdiagramm *Bild 6* zeigt, wurde in die Norm aufgenommen, um dem Steuergerät möglichst schnell einen Überblick über wichtige Zustände aller angeschlossenen Geräte zu verschaffen. Der Grundgedanke: Jedes Gerät sendet über eine ihm zugewiesene Bus-Datenleitung eine 1-Bit-Zustandsnachricht. Nimmt man einmal an, daß 8 Geräten im Bus-System je eine Datenleitung zugewiesen wurde, dann empfängt das Steuergerät bei einer Parallelabfrage ein Byte mit den Zustandsbits der 8 Geräte. Dieses Byte wird im Gegensatz zu den Statusbytes nicht im Handshake-Verfahren übertragen. Das Steuergerät startet die Parallelabfrage durch gleichzeitiges Aktivieren der Busleitungen ATN und EOI. Die Geräte plazieren dann sofort ihre Zustandsbits auf den Datenleitungen. Nach der Einschwingzeit kann das Steuergerät die Information lesen und verarbeiten. Vorab sei noch erwähnt, daß mehreren Geräten dieselbe Datenleitung zugewiesen werden kann. Zur eindeutigen Identifizierung eines Geräts sind dann allerdings dynamische Zuordnungen und weitere Abfragen vonnöten.

Die PP-Funktion ist, ebenso wie die T-Funktion (Bild 3), in Form zweier Teildiagramme beschrieben. Ihr Ruhezustand müßte demnach korrekt mit (PPIS, PUCS) bezeichnet werden. Im folgenden wird diese etwas umständliche Darstellungsweise vermieden, da bei allen Aktivitäten bezüglich eines Teildiagramms der Zustand des anderen Teildiagramms bestimmbar ist. Zunächst muß in einer Konfigurationsphase den PP-Funktionen der einzelnen Geräte mitgeteilt werden, auf welcher Datenleitung sie ihr Zustandsbit senden sollen. Der vom Steuergerät ausgesandte Befehl PPC („Parallel Poll Configure") bringt die PP-Funktionen derjenigen Geräte in den Zustand PACS,

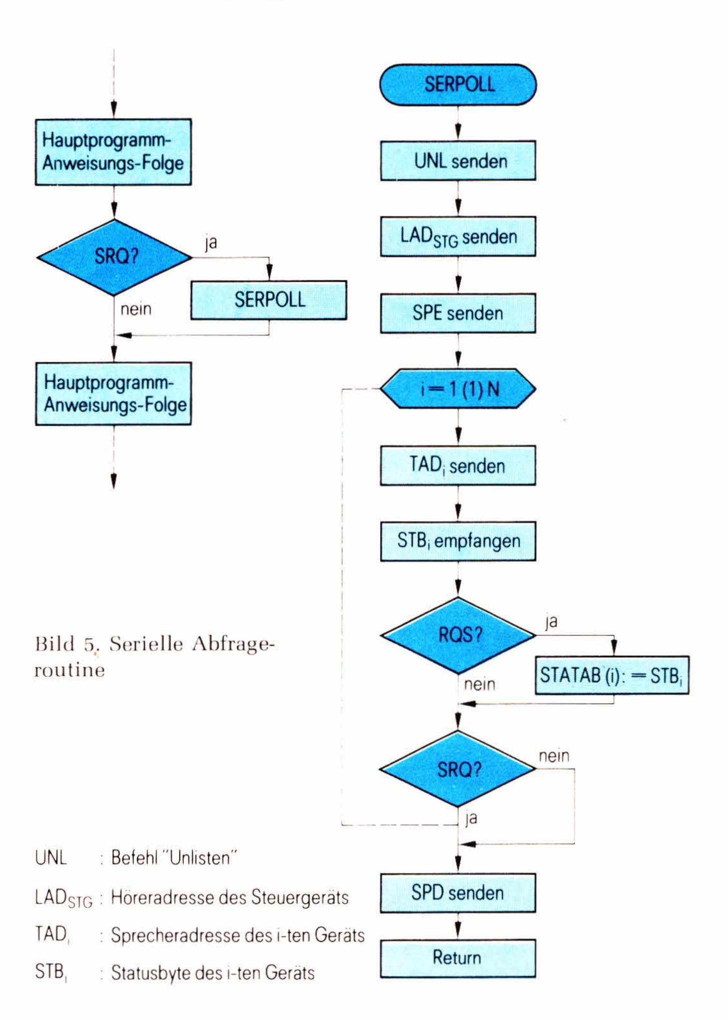


Bild 5. Serielle Abfrageroutine

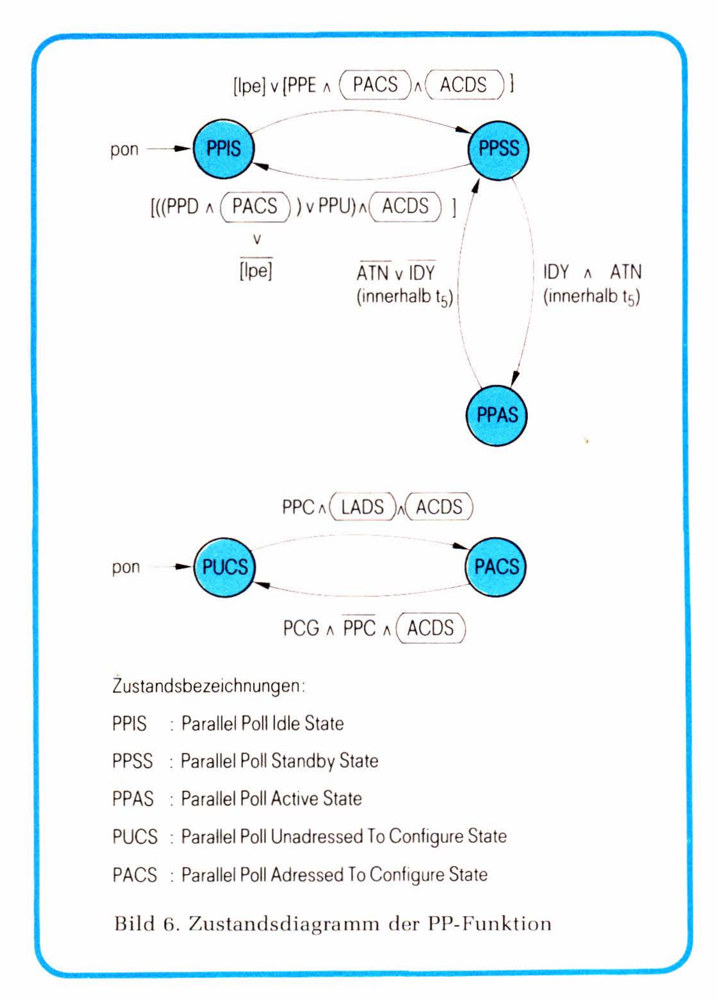


Bild 6. Zustandsdiagramm der PP-Funktion

die als Hörer adressiert sind. PPC ist also ein „adressierter Befehl". Jede ausgewählte PP-Funktion ist nun bereit für den Empfang der Programmierinformation. Diese wird mit einem PPE-Befehl („Parallel Poll Enable") übertragen. PPE hat die Form $X110SP_3P_2P_1$. Die drei niedrigstwertigen Bits $P_3 ... P_1$ bezeichnen dabei die Nummer der zugeordneten Datenleitung, die von der PP-Funktion gespeichert werden muß: das Aktivierungsbit S legt zusammen mit dem Zustandsbit, das die Gerätefunktionen als interne Nachricht ist („individuell status bit") an die PP-Funktion senden, fest, ob als PP-Antwort eine „1" (ist≡S) oder eine „0" (ist≢S) gesendet wird. Im Fall der Zuordnung mehrerer Geräte zu derselben Datenleitung lassen sich damit „verdrahtete" UND- bzw. ODER-Verknüpfungen programmieren. S wird ebenfalls gespeichert. Beispiel: PPE = 01101010 ordnet die Bus-Datenleitung DIO3 zu (Bezeichnungsweise in der Norm: DIO1...DIO8); ist=„1" führt zur PP-Antwort „1".

Die PP-Funktion ist nun im Zustand PPSS und damit bereit, auf eine Parallelabfrage zu antworten. Der Zustand PACS, der nur für die Konfigurationsphase von Bedeutung war, wird verlassen, wenn vom Steuergerät eine Adresse, ein Universalbefehl oder ein adressierter Befehl übertragen wird. Dies vereinfacht die Konfigurationsphase: Sobald ein neues Gerät per Höreradresse angewählt wird, geht die PP-Funktion des Vorgängergeräts zurück nach PUCS.

Die Bedingung

$$\text{IDY} \wedge \text{ATN} \qquad (3)$$

für den Übergang von PPSS nach PPAS ist erfüllt, wenn das Steuergerät die Busleitungen EOI(=IDY) und ATN aktiviert. Alle vorbereiteten PP-Funktionen antworten dann innerhalb von 200 ns auf „ihrer" Datenleitung. Das Steuergerät beendet die Parallelabfrage, indem es Bedingung (3) nicht länger erfüllt: die PP-Funktionen gehen wiederum nach höchstens 200 ns zurück in den Zustand PPSS. Wie bereits angedeutet, ist es bisweilen notwendig, ein Gerät von der Parallelabfrage auszuschließen bzw. ihm eine andere Antwortleitung zuzuordnen. Dies läßt sich durch geschickte Anwendung der Befehle PPD („Parallel Poll Disable") und PPE erreichen: Die PP-Funktion des betreffenden Geräts muß sich im Zustand PACS befinden; PPD desaktiviert zunächst das Gerät, macht aber auch den Weg frei für die Neuprogrammierung mittels eines PPE-Befehls. An dieser Stelle ist anzumerken, daß die Norm nichts darüber aussagt, wann und wie die PP-Funktion die Programmierinformation von PPE-Befehlen speichert. Eingedenk der 16 möglichen PPE-Befehle müßte das PP-Zustandsdiagramm hierzu anstelle von PPSS und PPAS die 32 Zustände $PPSS_i$ und $PPAS_i$ ($\emptyset \leqq i \leqq 15$) aufweisen. Deren Übergänge und zugeordnete Ausgaben erlaubten dann eine eindeutige Formulierung.

Bild 7 skizziert, wie man jedem von N Geräten eine der 8 Bus-Datenleitungen zuordnen kann: Ein PPU-Befehl hebt jede bestehende Zuordnung auf. Alle N Geräte werden daraufhin nacheinander angewählt und erhalten gemäß einer Vorschrift Datenleitung und Steuerbit zugewiesen. Der Befehl UNL bringt dabei die PP-Funktionen in den Zustand PUCS zurück.

Auf die von der Norm vorgesehene „Teilausrüstung PP2" der PP-Funktion, die eine interne Nachricht lpe (local poll enable) zur Eigeneinstellung vorsieht und wenig flexibel ist, soll hier nicht näher eingegangen werden.

4 Ein Konzept für leistungsfähige IEC-Bus-Geräte

Wurden in den beiden vorangehenden Abschnitten Schnittstellenfunktionen für den Umgang mit Zustandsinformation vorgestellt, so ist nunmehr das Augenmerk auf die Gerätefunktionen (Bild 1) IEC-Bus-kompatibler Geräte zu richten. Die Gerätefunktionen realisieren das Arbeitsprinzip eines Geräts (Digitalvoltmeter, Drucker, Floppy-Disk-Speicher) und bestimmen, natürlich zusammen mit den Schnittstellenfunktionen, dessen Leistungsfähigkeit und Bedienungskomfort. Insofern unterscheiden sie sich von Gerätetyp zu Gerätetyp. Im folgenden soll trotzdem anhand von Beispielen der Versuch gemacht werden, einige gemeinsame Verhaltensmerkmale von IEC-Bus-Geräten herauszuarbeiten. Beispiele hierfür soll der Einsatz der IEC-Bus-Schnittstelle in kleinen Prozeßrechensystemen liefern (ein zweifellos anspruchsvolles und von den „Erfindern" der Norm vielleicht gar nicht beabsichtigtes Unterfangen [8]): Das wachsende Angebot von leistungsfähigen Minicomputern mit IEC-Bus-Schnittstelle [9] und die Vermutung, daß durch konsequente Ausnutzung einer existierenden Schnittstellennorm die Entstehung immer neuer Va-

rianten verhindert werden kann, bestärken diesen Vorsatz.

4.1 Der „geräteinterne" Prozeß

Für jedes Gerät, das Bestandteil eines größeren Systems ist, läßt sich ein geräteinterner Prozeß formulieren, der im wesentlichen zwei Aufgabenbereiche bearbeitet.

- Gerätespezifische Tätigkeiten: Bei einem Digitalvoltmeter heißen diese z. B. „Steuerung des Meßvorganges" und „Formatierung des Meßwerts"; bei einem Floppy-Disk-Speicher sind u. a. zu unterscheiden „Spurpositionierung", „Sektorsuchen", „Schreiben", „Lesen".
- Kommunikation mit dem System: Bei IEC-Bus-Geräten ist eine der hier einzuordnenden Aufgaben die Erkennung bzw. Erzeugung des vereinbarten Datenformats. Für die syntaktische Struktur von Zahlenwerten gibt es Empfehlungen [10]; darüber hinaus können für jedes Gerät (d. h. für dessen geräteinternen Prozeß) individuelle Vereinbarungen getroffen werden. Einem Netzgerät wird beispielsweise der Zahlenwert der einzustellenden Spannung mit dem ASCII-Zeichen „U" angekündigt, und „I" leitet den Zahlenwert für die Strombegrenzung ein. Ein Floppy-Disk-Speicher, der jeweils einen physikalischen Sektor behandeln kann, benötigt Operationscode, Spur- und Sektoradresse in einem festgelegten Format [11]. Die Kommunikation mit dem System läßt sich aus der Sicht des geräteinternen Prozesses auf ein Wechselspiel von Auftragsempfang und Auftragsbearbeitung zurückführen. Das Digitalvoltmeter erhält einen Meßauftrag und liefert einen Meßwert ab, der Drucker erhält eine Zeichenkette und druckt sie, der Floppy-Disk-Speicher erhält Lesebefehl und Adresse und bietet daraufhin die gelesene Information an. Ein Funktionsgenerator erhält detaillierte Information über Form, Amplitude und Frequenz und stellt das Gewünschte an den Ausgangsklemmen bereit – bis zum nächsten Auftrag.

4.2 Gerätezustandsmeldungen

Die stürmische Ausbreitung der IEC-Bus-Schnittstelle in den letzten Jahren hat zu Gerätesystemen geführt, in denen die Steuereinheit (meist ein BASIC-interpretierender Tischrechner) fast nichts über die Zustände der geräteinternen Prozesse weiß. Der Auslastungsgrad des Übertragungssystems ist dabei gering: Es wird von zentraler Stelle für Übertragungen vorbereitet, über deren Zustandekommen dann die beteiligten Geräte dezentral entscheiden. Erhebliche Wartezeiten sind die Folge, in denen das Gesamtsystem untätig ist. Typisches Beispiel: Ein Programm, das auf dem Prozessor der Steuereinheit läuft, stößt ein hochgenaues, d. h. langsames Digitalvoltmeter zu einer Messung an, bereitet das Bus-System für die Übertragung des Meßwertes vor, und wartet dann, bis diese zustandegekommen ist. Fehlerhafte Gerätezustände, hervorgerufen durch falsche Programmierung oder unvorhergesehene Betriebsbedingungen, sind in überschaubaren oder unkritischen Anwendungen vermeidbar oder können toleriert werden. Angesichts steigender Kosten für Software und Systementwurf ist die bisherige Entwicklung bei IEC-Bus-Systemen allerdings verständlich: Sie entspricht den Anforderungen der Prüf- und Laborautomatisation.

Prozeßrechensysteme hingegen sind vom Prinzip her niemals untätig. Innerhalb vorgegebener Zeitspannen, die natürlich möglichst kurz sein sollen, müssen Überwachungs-, Steuerungs- und Regelungsaufgaben durchgeführt werden; zwischen den Einzelprozessen muß „umgeschaltet" werden nach Maßgabe ihrer Prioritäten und der von ihnen benötigten Betriebsmittel. Dies ist nicht der Platz, um näher auf die Grundlagen der Prozeßrechentechnik einzugehen. Festzustellen bleibt nur, daß in derartigen Systemen zu jedem Zeitpunkt ein vollständiger Überblick über die Verfügbarkeit der Betriebsmittel vorhanden sein muß. Im obigen Beispiel wird deshalb während der Meßzeit des Digitalvoltmeters der Prozessor der Steuereinheit einem anderen Teilprozeß zugewiesen werden. Der Teilprozeß, der den DVM-Meßwert verarbeitet, darf erst dann fortgesetzt werden, wenn der Meßwert zur Übertragung bereitsteht. Das Digitalvoltmeter muß also die Beendigung seines Auftrages selbständig („aktiv") einem Verwaltungsprogramm melden, das darüber entscheidet, welcher Einzelprozeß weiterlaufen darf.

Ein geräteinterner Prozeß bedient sich dabei der Schnittstellenfunktion SR („Service Request"). Bereitstellung des Statusbytes und Bedienungsanforderung über die SRQ-Busleitung erfolgen immer dann, wenn mit dem System kommuniziert werden soll, d. h.

- wenn ein Geräteauftrag abgeschlossen wurde (Beispiel DVM: „Meßwert bereit zur Übertragung" oder „Meßbereichsüberschreitung, Meßwert ungültig"), und
- wenn ein Geräteauftrag nicht zu Ende geführt werden kann (Beispiel Floppy-Disk-Speicher: „Sektor nicht gefunden"; Beispiel Funktionsgenerator: „Unerlaubter Zahlenwert für Frequenzeinstellung").

In beiden Fällen müssen eindeutig die Aktionen spezifiziert sein, die

- der geräteinterne Prozeß vom System erwartet (Beispiel DVM: „Meßwertübernahme" bzw. „Erteilung eines neuen Auftrags"; aber u. U. auch: „Gerät nicht mehr ansprechen" bei Ausfall wichtiger Gerätefunktionen), sowie
- vom Gerät selbst durchgeführt wurden (Beispiel Netzgerät: „Selbständiges Abschalten der Ausgangsspannung bei Überschreitung der Strombegrenzung").

4.3 Schnelle Bearbeitung von Bedienungsanforderungen

Geräte, die oft und präzise über ihren Zustand Auskunft geben, führen natürlich in Einfach-Systemen zu einem leicht erhöhten Programmieraufwand für jede Übertragung, denn alle Bedienungsanforderungen müssen bearbeitet werden, weil sie die SRQ-Busleitung belegen. Mit Hilfe weitverbreiteter höherer Sprachen (z. B. BASIC) ist dieser Zusatzaufwand leicht zu bewältigen.

Prozeßrechensysteme erfordern die schnelle und für den aktuell ablaufenden Einzelprozeß transparente Entgegennahme der Gerätemeldungen. Eine Geräteeigenschaft, die diesen Vorgang unterstützt, wird im folgenden beschrieben. Zunächst ist zu fordern, daß jede Bedienungsanforderung auf der SRQ-Busleitung eine Unterbrechung („Interrupt“) im Steuergerät auslöst. Als Bedienungsroutine („Interrupt Service Routine“) wird daraufhin eine serielle Abfrage gestartet, die herausfinden muß, welches Gerät Bedienung angefordert hat. (Der zweite Schritt, die geeignete Reaktion auf die Zustandsinformation des anfordernden Geräts, ist in hohem Maße systemabhängig und soll hier nicht Gegenstand sein.) In einem IEC-Bus-System mit mehr als drei intensiv benützten Geräten, die alle in der vorgeschlagenen Weise Bedienungsanforderungen absetzen, stellen die häufigen zeitaufwendigen Abfragen bereits eine starke Belastung der Steuereinheit dar. Diesem Nachteil ist zu begegnen, indem die PP-Schnittstellenfunktionen der Geräte zur Identifikation herangezogen werden. Die internen Nachrichten rsv (an die SR-Funktion) und ist (an die PP-Funktion) werden zu diesem Zweck von den Gerätefunktionen zu denselben Zeitpunkten gesetzt bzw. rückgesetzt. Bei geeigneter Programmierung der einzelnen PP-Funktionen in der Konfigurationsphase (Bild 7) läuft eine schnelle serielle Abfrage von bis zu 8 Geräten wie folgt ab: Zuerst wird eine Parallelabfrage durchgeführt. Das hierbei gewonnene Byte bezeichnet durch eine Eins in einer Bitposition ein Gerät, das Bedienung gefordert hat. Daraufhin wird gezielt das Statusbyte des markierten Geräts geholt, wobei das RQS-Bit zur Kontrolle herangezogen werden kann. In einem System mit mehr als 8 Geräten werden einige Bus-Datenleitungen 2 Geräten zugeordnet sein („Verdrahtetes ODER“). Unter Umständen müssen dann 2 Statusbytes geholt werden, um das anfordernde Gerät ausfindig zu machen.

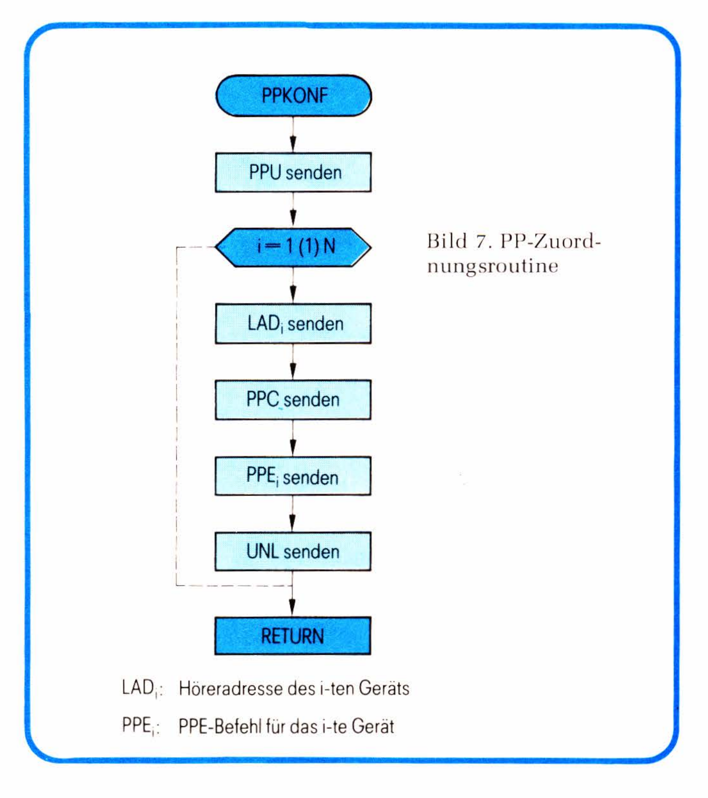


Bild 7. PP-Zuordnungsroutine

5 Ein universeller Sprecher/Status-Modul

Bild 8 zeigt eine Schaltung, die einem IEC-Bus-Gerät die Fähigkeit gibt, Datenbytes und, auf Anforderung, zu beliebigen Zeitpunkten auch Statusbytes zu senden, letzteres mit der vorgeschlagenen Unterstützung durch die PP-Funktion. Drei Datenquellen „Datenregister“, „Statusregister“ und „Parallel-Poll-Antwort“ werden abhängig von den Zuständen der 4 Schnittstellenfunktionen SH („Handshake-Quelle“, „Source Handshake“), T, SR und PP auf den Bus geschaltet. Außerdem werden die internen Nachrichten an diese Schnittstellenfunktionen erzeugt. Um das Prinzip hervorzuheben, sind in Bild 8 nur die Ein- und Ausgänge der Schnittstellenfunktionen dargestellt, die für den Ablauf wichtig sind. Für die Signalbezeichnungen gilt: Der Ausgang xxxs ist genau dann aktiv, wenn sich die Schnittstellenfunktion im Zustand XXXS befindet.

In Geräten, deren Gerätefunktionen durch Mikrocomputer gesteuert werden, sind „Datenregister“ und „Statusregister“ zweckmäßigerweise mittels sogenannter paralleler Schnittstellenbausteine zu realisieren, die Mitglieder der meisten Mikrocomputerfamilien sind (Z80: PIO; 6800: PIA 6820; 8080/85: 8255). Hier soll von einer allgemeinen Ablaufsteuerung für den geräteinternen Prozeß ausgegangen werden; dabei bewirkt ein Signal DATUMSTROBE die Übernahme eines Bytes vom „Gerätebus“ DB1...DB8 in das Datenregister und setzt das Daten-Flipflop. Wenn gerade kein Übertragungszyklus für ein Statusbyte im Gange ist, $\overline{\text{spas}}$ also „1“-Pegel führt, wird sofort über NBADATUM die interne Nachricht nba („new byte available“) an die SH-Funktion erzeugt. Während der Übertragung des Datenbytes gelangt die SH-Funktion in den Zustand STRS („Source Transfer State“).

$$\text{STRSDATUM} = \text{strs} \wedge \overline{\text{spas}} \quad (4)$$

setzt Daten-FF und nba zurück und meldet das Ende der Übertragung an die Ablaufsteuerung.

Bedienungsanforderungen werden bei jedem Laden des Statusregisters erzeugt. STATUSSTROBE bewirkt die Übernahme und setzt das Bedienungsanforderungs-FF. Das Signal rsvist geht, wie in 4.3 beschrieben, an die Funktionen SR und PP. Wenn die Übertragung des Statusbytes sichergestellt ist (Zustand APRS der SR-Funktion), wird rsvist zurückgenommen. Statusbyte-Übertragungen werden vom Steuergerät eingeleitet; während solcher Zyklen ist die T-Funktion im Zustand SPAS, und der geräteinterne Prozeß darf das Statusregister nicht neu laden. Zu diesem Zweck ist spas aus dem Modul herausgeführt: Es kann von der Ablaufsteuerung als Aufforderung interpretiert werden, zum nächstmöglichen Zeitpunkt anzuhalten und dann die Bereitmeldung STATUSSENDEN zu aktivieren, die über das Status-FF NBASTATUS und nba an die SH-Funktion erzeugt. (Denkt man sich die Ablaufsteuerung durch einen Mikrocomputer auf 8080-Basis realisiert, würde spas den HOLD-Eingang des Prozessors steuern; STATUSSENDEN wäre identisch mit dem Quittungssignal HLDA.)

$$\text{STRSSTATUS} = \text{strs} \wedge \text{spas} \quad (5)$$

setzt Status-FF und nba zurück und kann als Rückmeldung verwendet werden.

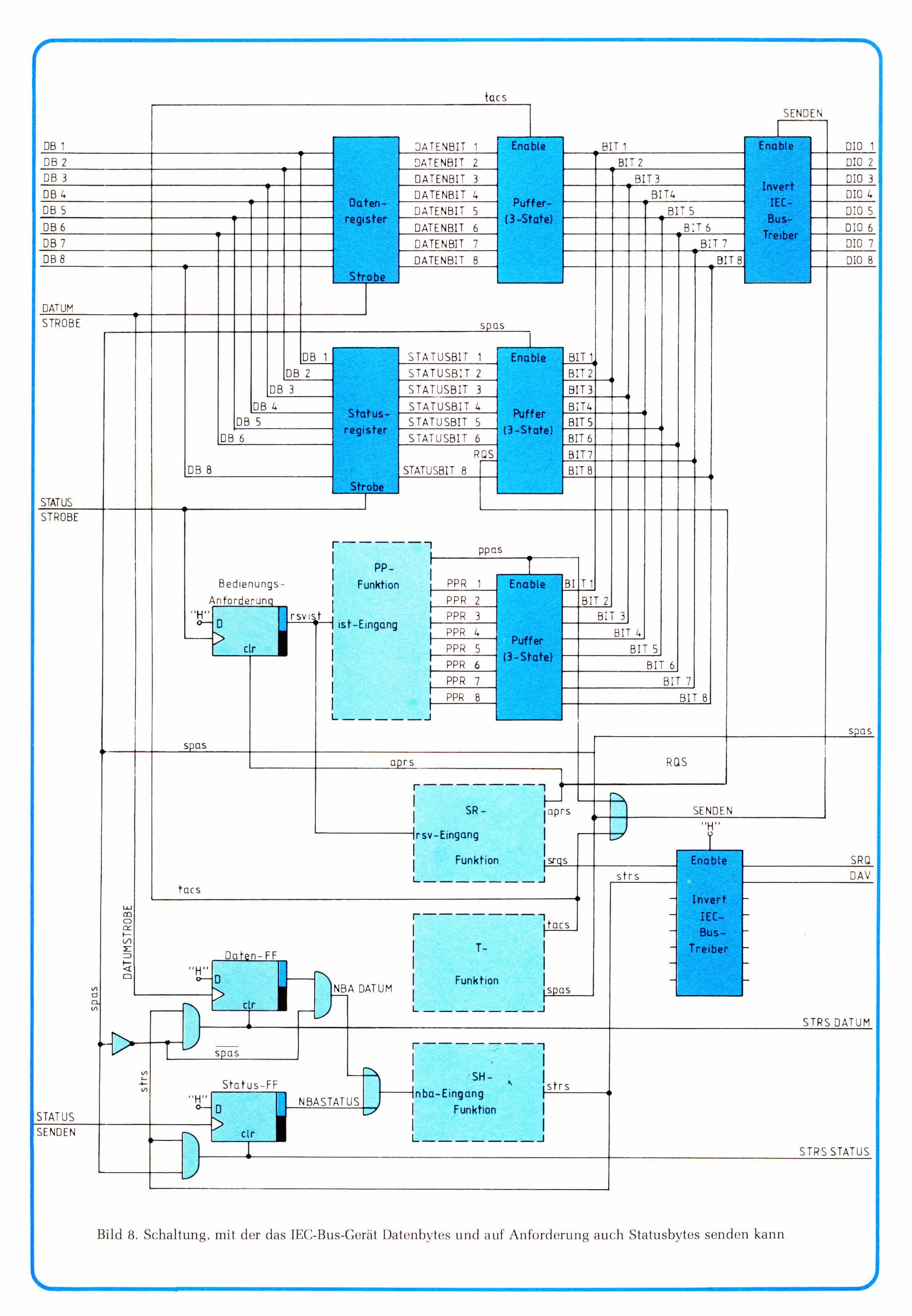


Bild 8. Schaltung, mit der das IEC-Bus-Gerät Datenbytes und auf Anforderung auch Statusbytes senden kann

Zwei problematische Fälle sind in der Schaltung berücksichtigt worden:

● Gerät bietet Datenbyte an (Daten-FF gesetzt, NBADATUM erzeugt nba); Steuergerät verlangt jedoch Statusbyte. Dann gilt $\overline{\text{spas}}$ =„0"; NBADATUM verschwindet, STATUSSENDEN setzt Status-FF und aktiviert NBASTATUS und nba. Wegen (4) und (5) wird im Verlauf der Statusbyte-Übertragung das Status-FF rückgesetzt. Das Daten-FF hingegen „merkt sich" weiterhin den Gerätewunsch nach einer Datenbyte-Übertragung und aktiviert, wenn wieder $\overline{\text{spas}}$ =„1" gilt, über NBADATUM die interne Nachricht nba.

● Statusbyte-Übertragungszyklus im Gang ($\overline{\text{spas}}$ =„0"); Gerät bietet Datenbyte an. Daten-FF wird zwar gesetzt, NBADATUM bleibt inaktiv. Während der Statusbyte-Übertragung wird wiederum nur das Status-FF rückgesetzt; nach der Übertragung aktiviert $\overline{\text{spas}}$ =„1" über NBADATUM die interne Nachricht nba. Damit kann die Statusbyte-Übertragung zu jedem Zeitpunkt im Ablauf des Geräteprozesses erfolgen.

6 Zusammenfassung

Gezeigt wurden Möglichkeiten, die beiden Schnittstellenfunktionen für „Bedienungsruf" und „Parallelabfrage", zu deren Ansteuerung die Norm kaum Hinweise gibt, sinnvoll zur Verbesserung des Systemverhaltens IEC-Bus-kompatibler Geräte einzusetzen. Am Lehrstuhl für Technische Elektronik der Universität Erlangen-Nürnberg wurden einige Geräte mit den genannten Eigenschaften ausgestattet. Der geringe zusätzliche Hardwareaufwand schlug dabei kaum zu Buch angesichts der Erleichterungen, die sich bei der Organisation komplexer und leistungsfähiger IEC-Bus-Systeme für den Systemprogrammierer ergaben. Erfahrungen wurden sowohl mit tischrechnergesteuerten Systemen als auch mit Mikro- und Minicomputer-„Steuergeräten" gesammelt. Der Anwendungsbereich der Geräte verschob sich nicht, wie zu befürchten stand, in Richtung auf anspruchsvolle Anwendungen, sondern erfuhr eine echte Erweiterung.

Der Verfasser dankt den Herren *Dr. H. Köhler* und *Dipl.-Ing. W. Schütz* für Unterweisung und Diskussion, Herrn *Ing.-grad. G. Unger* und *Fr. E. Munkert* für tatkräftige Unterstützung. Die Genannten sind Mitarbeiter am Lehrstuhl für Technische Elektronik der Universität Erlangen-Nürnberg (*Prof. Dr.-Ing. D. Seitzer*).

Literatur

1 Entwurf DIN IEC 66.22, Elektrische Meßtechnik, „Byteserielles bitparalleles Schnittstellensystem für programmierbare Meßgeräte", Beuth-Verlag, Berlin, 1976.

2 *Pannach, A.*: Interface-Entwicklung für den IEC-Bus. ELEKTRONIK 1975, H. 12, S. 61...64.

3 *Richter, M.*: Das Zustandsdiagramm und seine Anwendung beim IEC-Bus. ELEKTRONIK 1977, H. 2, S. 55...58, 71.

4 *Köhler, H.*: Interface-Moduln für den IEC-Bus. Elektronikpraxis 1977, H. 5, S. 7...12 und H. 6, S. 16...25.

5 Applikationsschrift der Firma Intersil: ASCII Bus Interface Logic Uses IM5200 FPLA.

6 *Schenk, Ch., Tietze, U., Nürnberger, H.*: Ein Mikrocomputerprogramm zur IEC-Bus-Steuerung. Elektronikpraxis 1977, H. 4, S. 10...12.

7 Datenblätter der Firmen Motorola (MC 68488) und Valvo (HEF 4738).

8 *McDermott, J.*: The IEEE 488 bus plays a major role in programmable instrument systems. Electronic Design 24, Nov. 1976, S. 76...80.

9 *Richert, U.*: IEC-Bus-Interface für Prozeßrechner. ELEKTRONIK 1976, H. 12, S. 58...62.

10 Supplement (Draft) to IEC-Publication „Interface systems for programmable measuring apparatus byte-serial bit-parallel". Section 7: Code and format conventions. IEC, TC66/WG 3 (Secretary) 16, May 1977.

11 *Kilgenstein, R.*: Einsatz eines Floppy-Disk-Speichers. Diplomarbeit, Lehrstuhl für Technische Elektronik, Universität Erlangen-Nürnberg, 1977.

Dipl.-Inf. Heinrich Dietsch wurde in Bamberg geboren. In Erlangen studierte er Informatik; seit 1976 ist er wissenschaftlicher Mitarbeiter am Lehrstuhl für Technische Elektronik der Universität Erlangen-Nürnberg. Hauptarbeitsgebiete: Mikrocomputerstrukturen, Bussysteme, Prozeßdatenverarbeitung.
Diensttelefon: (0 91 31) 85 78 70
ELEKTRONIK-Leser seit 1975



Dipl.-Ing. Hans-Heinrich Freytag

Zur Kompatibilität im IEC-Bus-System

Probleme lassen sich lösen – Erweiterung der Norm auf einheitliche Codierungen und Datenformate

Beim Zusammenfügen von programmierbaren Meßgeräten und Steuerrechnern zu einem automatischen Meß-System treten häufig Schwierigkeiten beim Datenaustausch auf, die auf das Nichtübereinstimmen der Daten-Strukturen und verwendeten Codierungen der Gerätenachrichten zurückzuführen sind. Was hier getan werden kann und was in bezug auf Normung unternommen wird, beleuchtet dieser Beitrag.

1 Grundsätzliches

Der Normung in diesem Bereich sind insgesamt Grenzen gesetzt, da die Vielfalt der Anwendungen des standardisierten Bus-Systems einer starken Vereinheitlichung des Datenaustauschs der angeschlossenen Geräte entgegensteht. Die mit der Ausarbeitung der Norm befaßten Gremien sahen es daher als vordringlich an, die funktionellen, elektrischen und mechanischen Eigenschaften der Datenschnittstelle in automatischen Meß-Systemen ausführlich zu beschreiben. Für geräteabhängige Daten hingegen, die anwendungsbezogen sind, wurden in der Norm DIN-IEC 625 Teil 1 [1] wie in der damit identischen IEC-Publikation 625-1, bzw. in der USA dem IEEE-Dokument 488-1978, keine Festlegungen getroffen. Es bestand aber kein Zweifel, daß ein Teil 2 erforderlich sein wird, der diese Lücke in der Normung des IEC-Bus-Systems ausfüllt. Im intensiven Kontakt zwischen Herstellern von Meßgeräten und Anwendern der automatischen Meßtechnik und unter Berücksichtigung von Normungen für den Datenaustausch auf benachbarten Gebieten war es möglich, zu Vereinbarungen über die im IEC-Bus-System zu verwendenden Codierungen und Datenformate für geräteabhängige Nachrichten zu kommen, die die Beschreibung der Eigenschaften der Datenbus-Schnittstelle ergänzen und abrunden. Künftig diese Vereinbarungen weitgehend zu befolgen, sollte ein Anliegen aller Hersteller sein. Der zeitliche Aufwand bei der Inbetriebnahme automatischer Meßeinrichtungen, der im wesentlichen durch Software-Anpassungsarbeiten bestimmt ist, ließe sich dann noch beträchtlich verringern.

Seit Oktober 1979 liegt Teil 2 der Norm DIN-IEC 625 mit dem Titel „Vereinbarungen über Codes und Datenformate" als DIN-Entwurf [2] vor. Die endgültige Ausgabe der Norm ist in Vorbereitung; sie stimmt inhaltlich mit der ebenfalls vorbereiteten IEC-Publikation 625-2 voll überein.

Aus *Bild 1* sind die Kommunikationswege und ihre Schnittstellen zu ersehen, wie sie in einem aus einem Steuergerät, einem Meßgerät als Datensenke (Hörer) und einem Meßgerät als Datenquelle (Sprecher) bestehenden IEC-Bus-System vorhanden sind. Teil 1 der Norm DIN-IEC-625 bezieht sich dabei auf die Nachrichten, die zum Durchführen der Kommunikation erforderlich sind, Teil 2 auf die Struktur der Kommunikation selbst, die den Nachrichtenaustausch zwischen den Gerätefunktionen betrifft.

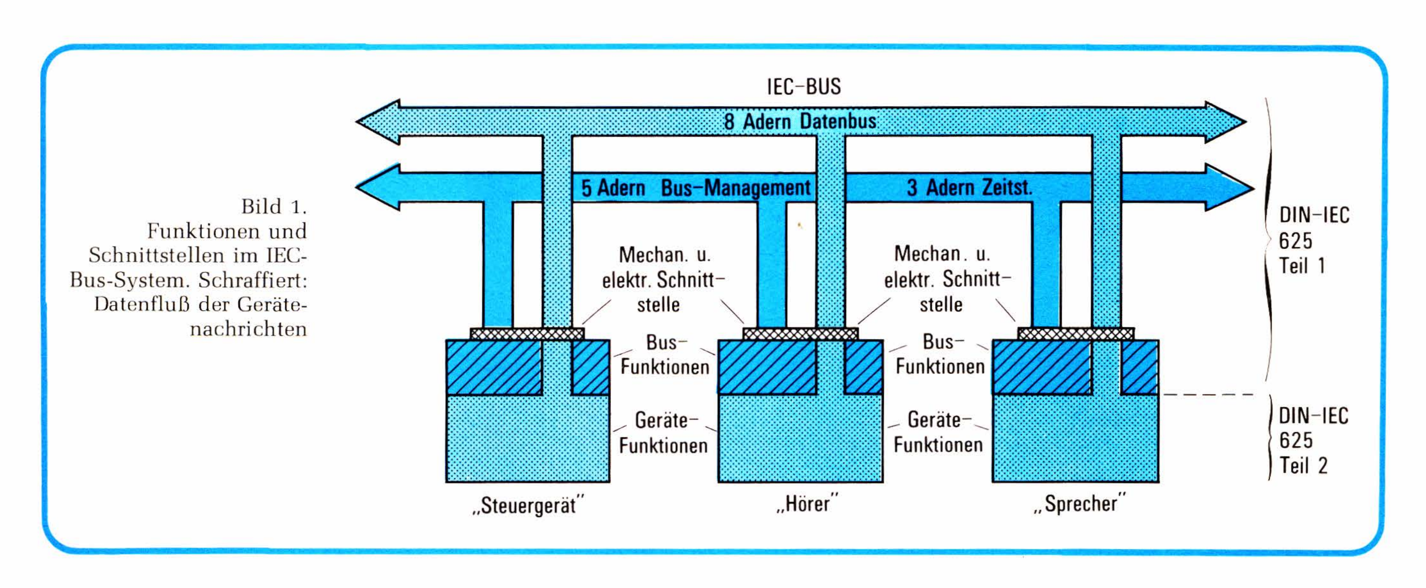


Bild 1. Funktionen und Schnittstellen im IEC-Bus-System. Schraffiert: Datenfluß der Gerätenachrichten

Der Begriff „Code“ bezieht sich auf das Bit-Muster geräteabhängiger Nachrichten, die auf den acht Datenleitungen des Systems als „Gerätenachrichten“ in byteserieller, bitparalleler Form und vorzugsweise im ISO-7-Bit-(ASCII-)Code übertragen werden. Der Begriff „Format“ umfaßt die Organisation der Daten-Bytes, die zum Austausch der Nachrichten benutzt werden. Diese Daten-Bytes repräsentieren im Zuge eines Programms die variablen Zeichenfolgen, die die eigentliche Information enthalten. Die Zeichenketten werden hierbei in Datenfelder unterteilt, um spezifische und erkennbare Komplexe, z. B. Zahlendarstellungen, innerhalb einer Nachricht wiederzugeben. Um die vielfältigen, und oft nicht vorhersehbaren Anwendungen des Schnittstellen-Systems IEC-625 nicht unnötig einzuschränken, läßt die Normung der Codes und Datenformate Alternativen zu. Sie gibt aber eindeutige Preferenzen, die in der Mehrzahl der Fälle befolgt werden können, ohne Nachteile im Datenaustausch in Kauf nehmen zu müssen, die aber die Systemkompatibilität entscheidend verbessern. Andererseits gibt die Norm durch die dargestellten Varianten dem Entwickler die Anregung, seine Konzeption nicht auf eine bestimmte Anwendung allein auszurichten, sondern auch andere Einsatzmöglichkeiten zu berücksichtigen.

Im folgenden soll ein Überblick über den Inhalt dieser Norm gegeben werden.

2 Datenarten und Datenfelder

Betrachtet man die in einem Bus-System ausgetauschten Nachrichten, so unterscheidet man Programmierdaten, z. B. von einem Steuergerät abgegebene Einstellbefehle in Richtung eines Meßgeräts und Meßdaten, die ein Meßergebnis enthalten, das dem Steuergerät zur weiteren Verarbeitung übergeben wird. Weitere Datenarten sind Zustandsdaten, die auf eine entsprechende Abfrage hin den augenblicklichen Betriebszustand eines Geräts melden oder Anzeigedaten als zusätzliche Information zu ausgegebenen Meßdaten, um die richtige Interpretation z. B. einer Anzeige zu ermöglichen. Die ersten beiden Datenarten lassen sich auf eine Grundstruktur der zusammenhängend übertragenen Zeichenketten zurückführen, deren Datenfelder sich in einen Kopfteil aus Buchstaben, einen Hauptteil mit den numerischen Angaben und den Trennungszeichen zur Begrenzung der einzelnen Zeichenketten, eines Datenblocks oder eines ganzen Datensatzes gliedern. Auf diese Grundstruktur soll zuerst eingegangen werden.

2.1 Kopfteil (Header)

Für den Kopfteil sind bevorzugt zwei Großbuchstaben zu verwenden, die die nachfolgenden numerischen Angaben näher kennzeichnen oder bewerten. Als Alternativen sind einer oder mehr als zwei Buchstaben, auch in Kleinschreibung, oder für Programmierdaten auch Ziffern, Satzzeichen oder Vorzeichen zugelassen. Werden mit Rücksicht auf den Empfänger feste Feldlängen vorgeschrieben, so können die Zeichenketten mit Zwischenraumzeichen (Δ) aufgefüllt werden, das erste Zeichen soll jedoch in jedem Fall ein Buchstabe sein, um das Kopfteil als solches immer von den weiteren Zeichenfolgen unterscheiden zu können. Meßdaten benötigen allerdings nicht immer ein Kopfteil, es ist daher für diese nicht zwingend vorgeschrieben.

2.2 Hauptteil (Body, numeric)

Das Hauptteil enthält die wesentliche Information, die Zahlenangaben. Für die Darstellung eines Meßergebnisses sind diese fraglos unerläßlich und daher zwingend vorgeschrieben. Man kann bei Dezimalzahlen unter drei Formen wählen: der Darstellung mit ganzen Zahlen, der mit einem je nach Stellenwert veränderlichen Dezimalpunkt und der exponentiellen Darstellung, die entweder mit festem oder veränderlichem Dezimalpunkt zu versehen ist. Einschränkungen betreffen die Verwendung der Vor- und der Zwischenraumzeichen. Der numerische Wert Null darf z. B. nur mit einem positiven Vorzeichen oder einem Zwischenraumzeichen in Verbindung gebracht werden. Die Darstellung ohne Zwischenraumzeichen ist jedoch vorzuziehen. Dem Dezimalpunkt soll immer eine Ziffer, auch eine Null, vorangestellt werden. Die Exponentialdarstellung kann entweder in der Mantisse ganze Zahlen enthalten, wobei der Exponent dem Stellenwert angepaßt wird oder es können beim Exponent Vielfache von drei verwendet werden, wobei der Dezimalpunkt in der Mantisse variiert wird. In die Ziffernfolge eingefügte oder nachfolgende Zwischenräume sind in allen drei Fällen nicht zugelassen. Nur bei Programmierdaten können Buchstaben als Anhängsel zur Ziffernfolge verwendet werden, um Zahlenangaben durch Multiplikatoren oder Divisoren, wie z. B. MA (Mega), K (Kilo), M (Milli), U (Mikro), oder N (Nano), [3] zu bewerten.

2.3 Trennungszeichenteil (Separator, delimiter)

Die Verwendung der richtigen Trennungszeichen ist meist ausschlaggebend für die Kompatibilität des Nachrichtenaustausches in einem System. Bei der Zusammenstellung von Geräten zu einem programmierbaren Meß-System ist daher besonderes Augenmerk auf die in den Meßgeräten und den Daten-Ausgabegeräten implementierten Endezeichen zu richten. Man unterscheidet zwischen Zeichen zur Trennung von Zeichenketten, von Blöcken und von Sätzen, die wiederum aus mehreren Blöcken bestehen. Nach Abgabe eines Satzes geht das sendende Gerät, der Sprecher, immer vom aktiven Sprecherzustand in den Ruhezustand. Dies wird im Datenverkehr über den Bus durch Setzen des Ende-Signals auf der Leitung EOI parallel zum letzten Datenbyte bewirkt. Zugelassen, aber wegen der Verwendung dieses Zeichens als Steuerzeichen im bit-seriellen Datenverkehr künftig nicht mehr zu bevorzugen, ist alternativ der ISO-7-Bit-Code ETX. Durch die Formatanweisungen an Zeilendrucker sind die Zeichen CR (Wagenrücklauf) auf LF (Zeilenvorschub) bereits bekannt. Sie können ebenso wie der ISO-7-Bit-Code ETB als Blockendezeichen verwendet werden. In Zukunft sollte sich aber das Endezeichen NL (Neue Zeile) hierfür durchsetzen, es ist mit dem

ISO-7-Bit-Code für LF identisch. Die kleinste Einheit, die Zeichenkette, wird durch ein Komma oder alternativ durch ein Semikolon beendet. Hiermit können beispielsweise mehrere Variable (z. B. einzustellende Frequenzen) oder Variablen-Paare (Amplitude und Phase) sowohl bei Meßdaten wie bei Programmierdaten voneinander getrennt werden. Das Endezeichen höherer Ordnung sollte immer das der niederen Ordnung ersetzen, Sequenzen von Endezeichen sind nicht erlaubt. Nach einem Blockendezeichen kann die Sprecherfunktion der Datenquelle in den Ruhezustand übergehen oder zumindest in den adressierten Zustand, wenn erwartungsgemäß zu einem späteren Zeitpunkt weitere Daten abgegeben werden sollen. Nach einem Satzendezeichen ist es üblich, aber nicht zwingend, daß der aktive Sprecherzustand an ein anderes, adressiertes Gerät übergeht. Es sei an dieser Stelle vermerkt, daß, entgegen der Festlegung im Entwurf DIN-IEC 625 Teil 2, Oktober 1979, nunmehr das Byte NL als Blockendezeichen zu bevorzugen ist. In *Tabelle 1* sind die zu verwendenden Trennungszeichen in der jetzt gültigen Fassung zur besseren Übersicht zusammengestellt. Hierin sind auch die international verwendeten anglo-amerikanischen Begriffe angegeben.

Tabelle 1. Codierung von Endezeichen

Feld	Endezeichenart	Codierung bevorzugt	Codierung zulässig
E_x	Zeichenkette (Sub-Record)	ISO-7-Bit-Code |,| (Komma)	ISO-7-Bit-Code |;| (Semikolon)
E_y	Datenblock (Record)	ISO-7-Bit-Code NL	ISO-7-Bit-Code ETB CR LF NL ∧ END
E_z	Satz (Record Terminator)	ISO-7-Bit-Code und END DAB ∧ END	ISO-7-Bit-Code ETX NL ∧ END

Kopft.	Hauptt.	E_x	Kopft.	Hauptt.	E_y	—	Kopft.	Hauptt.	E_z
← Zeichenkette →			← Zeichenkette →				← Zeichenkette →		
← Datenblock →							← Datenblock →		
← Satz →									

standsdiagramme [4] für die einzelnen Schnittstellen-Funktionen eingeführt worden sind.

3 Die Syntax von Meß- und Programmierdaten

Das Aneinanderreihen von Datenfeldern bei der byte-weisen Übertragung von Nachrichten muß syntaktischen Regeln folgen. Diese bestehen aus obligatorischen Festlegungen innerhalb der Norm, sowie aus Empfehlungen und zulässigen Alternativen. Um eine übersichtliche Darstellung bemüht, hat man zunehmend von Syntax-Diagrammen Gebrauch gemacht, wie sie in Sprachbeschreibungen verwendet werden. Sie erleichtern die Interpretation des Textes, so wie in der Norm DIN-IEC 625 Teil 1 aus diesem Grunde Zu-

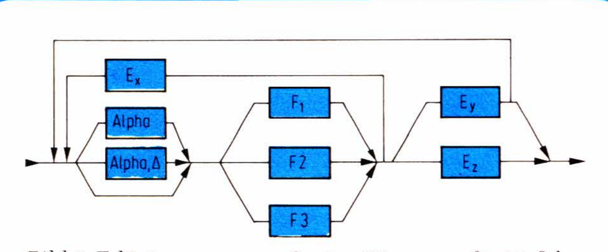


Bild 2. Erläuterungen zum Syntax-Diagramm für Meßdaten

- Das Kopfteil ist nicht obligatorisch. Es können Groß- oder Kleinbuchstaben (Alpha-Zeichen), ggf. bei festen Feldlängen auch Zwischenraumzeichen (△) verwendet werden.
- Das Hauptteil enthält Zahlenangaben (Numerische Zeichen). Die numerische Darstellungsform F1 gilt für ganze Zahlen, sie entspricht einer Schreibweise, bei der der Dezimalpunkt am Ende der Ziffernfolge angenommen, aber nicht übertragen wird. Die Darstellungsform F2 verwendet den Dezimalpunkt, dessen Position nach Stellenwert variiert. F3 ist die exponentielle Darstellungsform mit Mantisse, dem Buchstaben E und einem höchstens 2stelligen Exponenten mit Vorzeichen.
- Die Codierung der zulässigen Endezeichen ist in Tabelle 1 wiedergegeben.

3.1 Das Syntaxdiagramm für Meßdaten

Die in Abschnitt 2 erwähnten drei Typen von Datenfeldern setzen sich, wie in *Bild 2* gezeigt, zu einer syntaktischen Regeln folgenden Zeichenkette zusammen. Die rechteckigen Felder im Diagramm stellen die jeweiligen Datenfelder dar, deren Inhalt durch die in Abschnitt 2 wiedergegebenen Codierungsregeln bestimmt ist. Von links nach rechts weisende Pfeile zwischen den Feldelementen (Datenbytes) weisen auf die zeitliche Folge der übertragenen Nachrichten hin. Von rechts nach links weisende Pfeile sind Rückkopplungsschleifen, die die Möglichkeit ein- oder mehrmaliger Wiederholungen von Feldelementen anzeigen. Von links nach rechts führende Pfeile, die Feldelemente umgehen, zeigen an, daß diese Felder wahlweise verwendet werden können.

Aus Bild 2 ist ersichtlich, daß bei Meßdaten nur numerischer Hauptteil und das zugehörige Endezeichen obligatorisch sind. Das Trennungszeichen im Anschluß an eine Ziffernfolge kann sowohl das Block- wie das Satzendezeichen sein. Bei einer Folge numerischer Daten werden die einzelnen Elemente durch das Endezeichen für die Zeichenkette voneinander getrennt, die Länge der Nachricht ist jedoch nicht begrenzt.

Die folgenden Beispiele, die wie auch die weiteren Beispiele, der Norm entnommen sind, sollen die für Meßdaten zugelassenen Zeichenfolgen erläutern:

- Es soll eine Gleichspannung von 12,002 V im 10-V-Bereich eines Meßgerätes gemessen worden sein. Die Zahlenangabe in exponentieller Darstellung ist durch den Vermerk „Bereichsüberschreitung (OVERLOAD)“ und die Art des Meßwertes „Gleichspannung (DC)“ ergänzt. Die Nachricht lautet:
  OLDC+12002.E-03NL

Typische Alternativen sind

OLDC+12.002CRLF

oder

12002E-03ETB.

- Ein Frequenzzähler mit zwei Kanälen mißt Frequenzen von 4,23 MHz und 2,60 kHz. Die Nachricht lautet:

AFMAHZ4.23,BFKHZ2.60NL

Typische Alternativen wären:

AFMAHZ4.23,BFKHZ2.6|0∧END|
A4.23,B2.60ETX

|0∧END| besagt, daß parallel zum letzten Datenbyte das Zeichen END auf der Leitung EOI gesendet wird.

3.2 Das Syntaxdiagramm für Programmierdaten

In der gleichen Weise, wie das Syntaxdiagramm für die Meßdaten, ist auch dieses Diagramm zu interpretieren (*Bild 3*). Wie man sieht, hat man sich durch eine Anzahl von Darstellungsvarianten bemüht, den unterschiedlichen Anforderungen der Geräte bei deren Programmierung gerecht zu werden. Der eigentliche Inhalt der Datenfelder des Kopfteils und des Hauptteils ist geräteabhängig. Er entzieht sich weitgehend einer Normung. Hinsichtlich der Interpretation gelten die gleichen Regeln, wie sie unter 3.1 für das Syntaxdiagramm der Meßdaten vermerkt wurden. Im allgemeinen werden Buchstaben im Kopfteil der Zeichenkette mit einfachen numerischen Angaben zum Kennzeichnen von einzustellenden Bereichen verwendet werden, wie sie auch den Bedienelementen an der Frontplatte des Gerätes zugeordnet sind. Vor allem aber bei Frequenzeinstellungen wird man von den für den numerischen Teil festgelegten Zahlendarstellungen in vollem Umfang Gebrauch machen müssen. Trennungszeichen sind ggf. entbehrlich, wenn jeder Einstellbefehl einheitlich mit einem Buchstaben beginnt. Wenn jedoch eine Folge von Zahlenangaben erforderlich ist, und diesen kein Buchstabe zur näheren Kennzeichnung angehängt wird, sind Trennungszeichen zwischen den Zahlen einzufügen. Bei längeren Befehlsblöcken können sich aber auch Block- oder Satzendezeichen als notwendig erweisen.

Zur Illustration folgende Beispiele:

- Ein Voltmeter wird programmiert, Gleichspannungen (F0, FCTN0) im Bereich 10 V (R4, RNGE 4) nach Auslösung durch ein internes Triggersignal (T1M3, TRG1MODE 3) zu messen und den Meßwert nach Übernahme der Einstellbefehle (P, PROG) auszugeben. Die Zeichenkette lautet:

F0R4T1M3|P∧END| oder
FCTN0RNGE4TRIG1MODE3PROG

Aus *Bild 4* sind die Bitmuster ersichtlich, die sich im 1. Beispiel durch die Belegung der Datenleitungen mit Gerätenachrichten ergeben. Diese werden im Zustand 0 des Schnittstellensignals ATN übertragen. Das Steuergerät als „Sprecher" und das Voltmeter als „Hörer" werden über die entsprechenden Adressen C und V 1 adressiert. Zur Unterscheidung von den Gerätenachrichten wird hierbei ATN = 1 gesetzt.

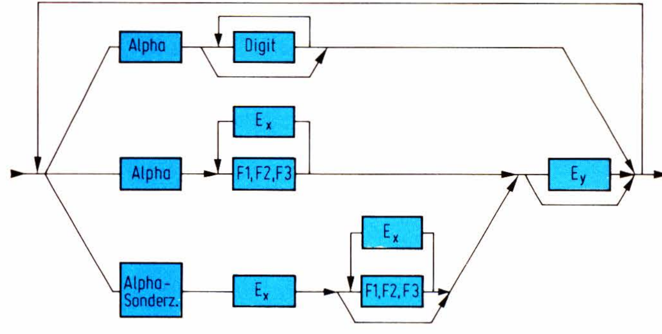


Bild 3. Erläuterungen zum Syntax-Diagramm für Programmierdaten:
- Das Kopfteil ist obligatorisch. Es können Groß- und Kleinbuchstaben (Alpha-Zeichen) und Sonderzeichen (Punkt, Komma, Semikolon, Plus- oder Minuszeichen) verwendet werden.
- Das Hauptteil enthält die numerische Angabe in Form von Ziffern, wobei die Darstellungsformen F1, F2 oder F3 zugelassen sind. Der Hauptteil ist wahlfrei.
- Zur Kennzeichnung von Bereichen können Alpha-Zeichen mit Ziffern (Digit) kombiniert werden.
- Größere Programmblöcke können durch das Endezeichen E_y unterteilt werden, wobei als Satz-Schluß-Zeichen dann E_z anstelle von E_y zu setzen ist. Die Kodierung der zulässigen Endezeichen ist in Tabellen wiedergegeben.

Verwendet man Trennungszeichen, so bildet sich folgende Zeichenkette:

F0,R4,T1,M3,P

- Ein Spektralanalysator wird programmiert, um 1000 Spannungsmessungen auszuführen und zu speichern, indem er die Betriebsart A3 für Speicherung und Anzeige verwendet. Die Meßwerte sollen bei einer Mittenfrequenz (DF) von 12,345 MHz einen Bereich (SP) von 2 MHz erfassen. Auf den Wobbelbefehl (TS) soll die Meßreihe gestartet werden. Die Zeichenkette lautet:

CF12.34 MAHZSP2000KHZTSA3 oder
CF12.34E+06HZSP2.00E+06HZTSA3

Mit Trennungszeichen alternativ:

CF12.34MAHZ,SP2000KHZ,TS,A3 oder
CF1234E-02MAHZSP2000E-03MAHZTSANL

4 Zustands- und Anzeigedaten

4.1 Struktur der Zustandsdaten

Im IEC-Bus-System ist es vorgesehen, daß sich einzelne Geräte über eine Bedienungsanforderung auf einer gesonderten Datenleitung melden können, wenn sie z. B. einen Meßvorgang beendet, eine Störung entdeckt oder auch eine Handbedienung notwendig haben [5]. Das Steuergerät entscheidet entsprechend der ihm eingegebenen Programmierung, ob der Ablauf des Programms unterbrochen wird und eine Abfrage der Geräte nach der Störungsursache erfolgen soll. Diese geschieht bei einer Serienabfrage nacheinander im Zustand „Serial Poll", Gerät für Gerät. Ein Gerät, das eine Bedienungsanforderung gesendet hat, wird über die auf die logische 1 gesetzte Datenleitung DIO7 erkannt. Die Datenleitungen der Bits 1...6 und Bit 8 können für die spezifische Zustandsnachricht verwendet werden. Hierbei kann über die Bits 1...4 frei verfügt werden, den Bits 5, 6 und 8 sind feste Bedeutungen zugeordnet. *Tabelle 2* gibt eine Übersicht über die vereinbarte Datenstruktur.

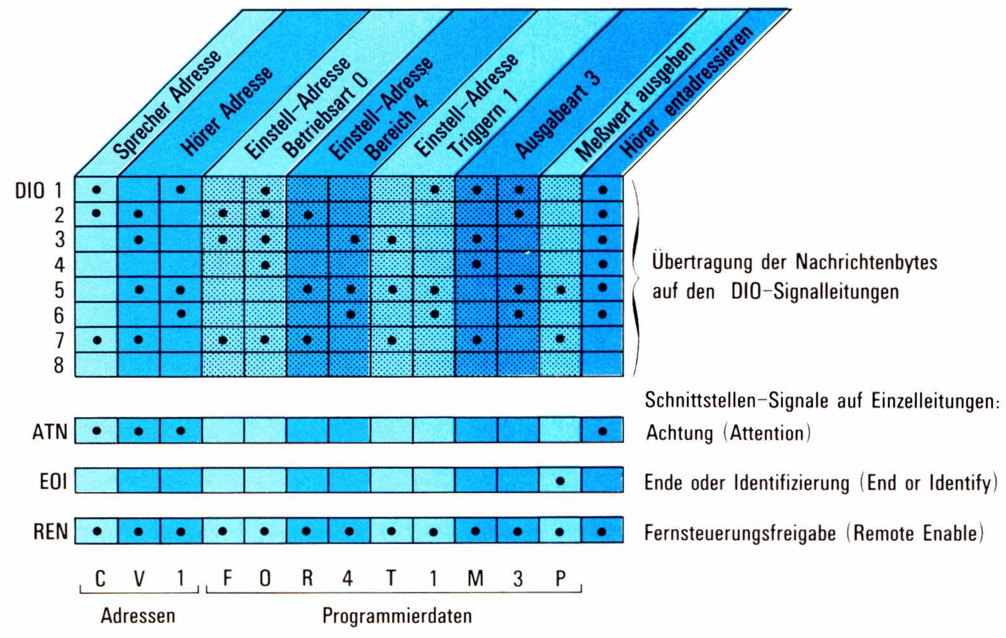


Bild 4. Bit-Muster der Nachrichtenbytes im ISO-7-Bit-Code am Beispiel der Programmierung eines Digitalvoltmeters. ● bedeutet: Leitung auf den logischen Zustand „1" gesetzt. Schraffiert: Datenbytes der Gerätenachrichten



Dipl.-Ing. Hans-Heinrich Freytag ist, nach Studium in Danzig und Dresden, seit 1947 Mitarbeiter der Siemens AG und gegenwärtig, im Rahmen allgemeiner Entwicklungsaufgaben, für die Vereinheitlichung rechnergestützter Meß- und Prüfverfahren im Bereich Weitverkehrssysteme tätig. Innerhalb der DKE, wie auch in der IEC, war er maßgebend an der Entstehung der IEC-Bus-Norm beteiligt. Von 1973 bis 1980 war er Vorsitzender der Working Group 3 des TC 66 der IEC, in der diese Norm durch internationale Zusammenarbeit entstanden ist.
Hobbys: Geschichte, Kunstgeschichte, Fotografie
ELEKTRONIK-Leser seit 1952

Wenn es jedoch für ein Gerät nur einen einzigen Anlaß für eine Bedienungsanforderung gibt, so ist auch das Setzen des Bit 7 ausreichend.

4.2 Struktur der Anzeigedaten

Die Anzeigedaten sollen im allgemeinen in ihrer Struktur den Meßdaten gleich sein. Da es sich um Ausgabedaten handelt, die Druckern oder Sichtgeräten zugeführt werden, muß darauf geachtet werden, daß nur Zeichen verwendet werden, die durch diese Geräte wiedergegeben werden können. Auf Satzzeichen, die mit Trennungszeichen verwechselt werden können, ist dabei besonders zu achten.

5 Codierungen und Codierungsregeln

Die Zeichen, mit denen Gerätenachrichten ausgetauscht werden, sind die Buchstaben (A|B|C|...), die Ziffern (0|1|2|3|...) und spezielle Symbole (+|–|, |;|...). Diese Zeichen sind nach der Vorschrift des ISO-7-Bit-Codes (DIN 66003) |6| verschlüsselt. Die Nachrichten enthalten entweder nur Ziffern oder Buchstaben und Ziffern kombiniert.

Es ist eine allgemeine Regel, daß das Bit, das die Zahlenangabe am wenigsten kennzeichnet, auf der Datenleitung mit der niedrigsten Nummer übertragen wird. Bei binärer Codierung ist dies die Datenleitung DIO1. Bei gepacktem BCD-Code sind es die Datenleitungen DIO1 und DIO5, die jeweils den Zahlenwert 2^0 repräsentieren. Die höhere Stelle wird in diesem Fall auf DIO5...DIO8 gelegt. DIO8 wird auf den logischen Wert 0 gesetzt, wenn diese Leitung nicht für die Zahlencodierung oder zur Paritätskontrolle verwendet wird.

Tabelle 2. Aufbau und Codierung von Zustandsdaten

Log. Zustand	RQS DIO7	DIO8	DIO6	DIO5	DIO4...1
1	Bedienung angefordert	erweitert	gestört	belegt	x x x x
0	Bedienung nicht angefordert	nicht erweitert	normal	bereit	x x x x

x = Gerätebedingte Codierung

Der ISO-7-Bit-Code kann auch für die Verschlüsselung von nichtdezimalen Zahlen herangezogen werden. Es ist jedoch darauf zu achten, daß die angesprochenen Geräte derartige Zahlen richtig interpretieren. Bei sedezimaler Zahlendarstellung betrifft dies auch die Buchstaben A...F bzw. die den Zeilen 10...15 des ISO-7-Bit-Code entsprechenden Sonderzeichen der Spalte 3 der Codetabelle. Mögliche Verwechslungen mit entsprechenden Trennungszeichen, wie dem Semikolon, müssen ausgeschlossen werden.

In dieser kurzgefaßten Darstellung konnten nicht sämtliche in der Norm enthaltenen Regelungen behandelt werden. Es ist daher bei der Beurteilung der Systemkompatibilität von IEC-Bus-Geräten unerläßlich, die Norm selbst zu Rate zu ziehen.

Literatur

|1| DIN-IEC-625 Teil 1. Ein byteserielles bitparalleles Schnittstellensystem für programmierbare Meßgeräte. Teil 1: Funktionelle, elektrische und mechanische Festlegungen, Anwendungen des Systems und Richtlinien für den Entwickler und Anwender (im Druck). Beuth Verlag, Berlin-Köln, 1980.

|2| DIN-IEC-625 Teil 2. Ein Schnittstellensystem für programmierbare Meßgeräte (byteseriell, bitparallel). Teil 2: Vereinbarungen über Codes und Datenformate (Entwurf Oktober 1979). Beuth Verlag, Berlin-Köln.

|3| DIN 66030, Darstellung der Einheitennamen mit beschränktem Schriftzeichenvorrat. Beuth Verlag, Berlin-Köln.

|4| *Richter, M.*: Das Zustandsdiagramm und seine Anwendung beim IEC-Bus. ELEKTRONIK 1977, H. 2, S. 55...58, 71.

|5| *Dietsch, H.*: Prinzip und Praxis der IEC-Bus-Funktionen „Bedienungsruf" und „Parallelabfrage". ELEKTRONIK 1978, H. 12, S. 89...96.

|6| DIN 66003, Informationsverarbeitung, 7-Bit-Code. Beuth Verlag, Berlin-Köln.

|7| *Freytag, H. H., Kriegeskotten-Thiede, A.*: Konzeption programmierbarer Meßgeräte mit IEC-Standard-Interface. Frequenz 30, 1976, H. 7, S. 190...195.

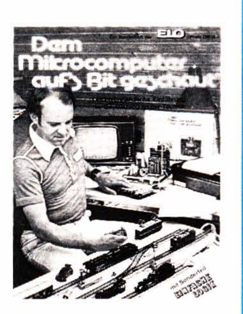
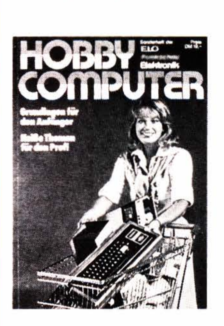
|8| *Frühauf, T.*: IEC-Bus-Ratgeber. ELEKTRONIK 1980, H. 8, S. 105...109; H. 9, S. 80...86.

Franzis-Sonderhefte

für alle Mikrocomputer-Interessierte

Welchen Weg der Mikrocomputer gehen wird, läßt sich noch nicht vorherbestimmen. Vor allem im Anwendungsbereich steht die Entwicklung erst am Anfang. Eines aber ist absehbar, jeder der auch nur am Rande mit Elektronik zu tun hat, wird davon berührt werden bzw. ist heute schon fest damit verbunden.

Mit diesen Sonde heften wollen wir jede Berufs- oder Hobb Elektroniker aktuell üb den neuesten Stand info mieren. Welches Heft für Sie als Info mationsquelle in Frage kommt, könne Sie aus nachstehender Tabelle ersehe

FUNKSCHAU-Sonderheft: Programme für Kleincomputer und Taschenrechner	ELO-Sonderheft: Dem Mikrocomputer auf's Bit geschaut	FUNKSCHAU-Sonderheft Mikrocomputer – eine Einführung in kleinen Schritten	Sonderheft der ELO, FUNKSCHAU und ELEKTRONIK Hobbycomputer 1	FUNKSCHAU-Sonderheft Hobbycomputer 2	ELEKTRONIK-Sonderheft 3 Mikroprozessoren
					
Für alle Computerbesitzer und solche, die es werden wollen	Anfänger	Anfänger	Anfänger und Fortgeschrittene	Fortgeschrittene	Industrielle Anwender
Programme für Taschenrechner (HP- und TI-Serien), für Basic-Computer und für die Mikroprozessoren Z-80 und 6502 auch in Maschinensprache Zahlreiche neue Beiträge.	Hardwareorientierte Einführung mit µP 2650. Sammlung der in der ELO bereits erschienenen gleichlautenden Beitragsreihe.	Hard- und Software-Grundlagen. Gesammelte Beiträge der in der FUNKSCHAU bereits erschienenen gleichnamigen Reihe.	Grundlagen, einfache Basic- und Maschinenprogramme. Überwiegend Beiträge, die bisher nirgendwo erschienen sind.	Basic- und Maschinenprogramme, Hardware-Tips. Ausschließlich exklusive Beiträge.	Programmierhilfen, Markttendenzen, 16-Bit-Prozessoren, Adressen von Herstellern. Überwiegend Beiträge aus der ELEKTRONIK
80 Seiten 14.80 DM	144 Seiten 18.– DM	68 Seiten 9.50 DM	136 Seiten 19.– DM	80 Seiten 14.– DM	136 Seiten 18.– DM

Wo gibt es diese Sonderhefte:

Sie erhalten diese Hefte bei allen Bahnhofsbuchhandlungen, Elektronik-Bauteilehändlern, größeren Buchhandlungen oder direkt beim Franzis-Verlag.

Bitte haben Sie Verständnis, daß der Verlag Einzelhefte aus organisatorischen Gründen nur gegen Vorauszahlung liefern kann. Wir bitten Sie, in diesem Fall als Bestellung den genannten Betrag plus 1.50 DM Porto auf unser Postscheckkonto München Nr. 813 75-809 mit genauer Nennung des jeweiligen Titels zu überweisen. Bitte vergessen Sie nicht, auf dem Zahlungsbeleg in Druckschrift Ihre volle Anschrift anzugeben. Sofort nach Eingang der Zahlung senden wir Ihnen das Heft zu.

Franzis-Verlag
Karlstraße 37, 8000 München 2
Telefon (0 89) 51 17-2 39

Die Hefte erhalten Sie auch in der Schweiz bei Verlag Thali AG, CH-6285 Hitzkirch

und in Österreich beim Fachbuch Center Erb, A-1061 Wien, Amerlingstraße 1.

Der vorliegende Aufsatz beschreibt ein Interface für die bekannte 16-Bit-Prozeßrechner-Familie PDP-11, das den Anschluß von Geräten mit IEC-Bus-Schnittstelle erlaubt. Sinngemäß können aus dieser Arbeit auch Denkanstöße zu Interface-Entwürfen für andere Rechner abgeleitet werden.

Ing. (grad.) Uwe Richert

IEC-Bus-Interface für Prozeßrechner

Beim Entwurf eines Interface muß davon ausgegangen werden, daß alle am IEC-Bus angeschlossenen Geräte gleichwertig sind. Sie unterscheiden sich lediglich dadurch, ob sie *alle* oder nur *einen Teil* der vom IEC-Komitee 66 definierten Interface-Funktionen verwenden. Innerhalb der einzelnen Interface-Funktionen existieren noch verschiedene Optionen (*Subsets*), die wahlweise verwendet werden können. Ein IEC-Bus-Interface kann somit exakt durch die einzelnen Funktionen und deren Subsets beschrieben werden (z. B. T 4).

Ein Prozeßrechner-Interface muß in jedem Fall die Control-Funktion (C) enthalten (Steuerfunktion), wenn es in der Lage sein soll, Befehle über den IEC-Bus zu senden und auf verschiedene Funktionen in den angeschlossenen Geräten zu reagieren (z. B. auf Service Request SR). Da zwischen dem IEC-Bus und dem angeschlossenen Prozeßrechner auch Daten in beiden Richtungen transferiert werden sollen, müssen außerdem die Talker- und Listener-Funktion mit den beiden Handshake-Funktionen (SH, AH) implementiert sein.

1 Interface-Funktionen

Gemäß der IEC-Spezifikation kann jedes am IEC-Bus angeschlossene Gerät einen Teil oder sämtliche der 10 definierten Interface-Funktionen umfassen. Zur Erinnerung sind diese Möglichkeiten nochmals in *Tabelle 1* zusammengefaßt. Wie daraus ersichtlich, benutzt das PDP-11-Interface nicht sämtliche Funktionen. Jedoch kann es durch die C-Funktion auf alle übrigen Funktionen reagieren, soweit diese in den angeschlossenen Geräten vorhanden sind.

Die Beschränkung auf nur sechs Funktionen hat praktische Gründe. Zum Beispiel ist es normalerweise überflüssig, ein Prozeßrechner-Interface mit der „*Remote Local*"-Funktion (RL) zu versehen. Andererseits muß es aber in der Lage sein, ein angeschlossenes Gerät in den „Remote"- oder „Local"-Zustand zu versetzen, was über die Control-Funktion (C) geschehen kann. Dies gilt prinzipiell auch für die Funktionen PP, DC und DT.

Tabelle 1. IEC-Bus-Interface-Funktionen

Abkürzung	Mögliche Funktionen	Verwendete Funktionen
SH	Source Handshake	SH 1
AH	Acceptor Handshake	AH 1
T	Talker	T 2
L	Listener	L 1
SR	Service Request	SR 1
RL	Remote Local	–
PP	Parallel Poll	–
DC	Device Clear	–
DT	Device Trigger	–
C	Control	C 1, 2, 3, 4, 5

2 Prozeßrechner-Schnittstelle

Die Schnittstelle zum Rechner wird über vier Register hergestellt, die per Programm gesetzt bzw. abgefragt werden können (*Tabelle 2*).

2.1 Control- und Interrupt-Register (CIR)

Das CIR dient zur Steuerung der Programm-Unterbrechungen, die verschiedene Ursachen haben können. Eine davon ist z. B. die Service-Request-Leitung (SRQ) des IEC-Bus. Zusätzlich besitzt das CIR spezielle Control- (d. h. Steuer-) Möglichkeiten, auf die später noch näher eingegangen wird.

2.2 State- und Message-Register (SMR)

Dieses Register ist das eigentliche „Herz" des Interface, da es einerseits zum Erzeugen von sogenannten „*Local Messages*" dient und andererseits einige markante Interface-Zustände anzeigt, wie z. B. „*Talker Active*" (TACS).

2.3 Input/Output-Register (IOR)

Das IOR besteht aus zwei Hälften (Bytes), eine für Datenausgabe, die andere für Dateneingabe. Wenn der Talker (TACS) oder Control (CACS) aktiv ist, muß das zu sendende Daten-Byte bzw. Befehls-Byte per Programm vom Rechner in den Ausgabepuffer geladen werden. Ist der Listener (LACS) aktiv, wird das empfangene Datenbyte per Programm aus dem Eingabepuffer abgeholt und in den Rechner übernommen.

2.4 Vector-Switch-Register (VSR)

Das VSR ist Unibus-spezifisch (*Unibus* ist die im Rechner PDP-11 benutzte Firmenbezeichnung für den rechnerinter-

Tabelle 2. Unibus-Register

Abkürzung	Register
CIR	Control- und Interrupt-Register
SMR	State- und Message-Register
IOR	Input/Output-Register
VSR	Vector-Switch-Register

nen Hauptbus) und zeigt die Adresse an, nach der das Programm bei Auftreten eines Programm-Interrupt verzweigt.

3 Aufbau

Das gesamte Interface ist auf einer gedruckten Platine von 20 x 39 cm^2 untergebracht und kann daher direkt in den Rechner hineingesteckt werden.

In der Blockschaltung (*Bild 1*) sind die Funktionsgruppen und die wichtigsten Signale dargestellt. Auf jede einzelne Funktion ausführlich einzugehen, würde zu weit führen, weshalb nachfolgend nur die wesentlichen Aspekte beleuchtet werden. Die in *Bild 1* benutzten Signalbezeichnungen entsprechen weitgehend der IEC-Spezifikation, soweit sie diese betreffen.

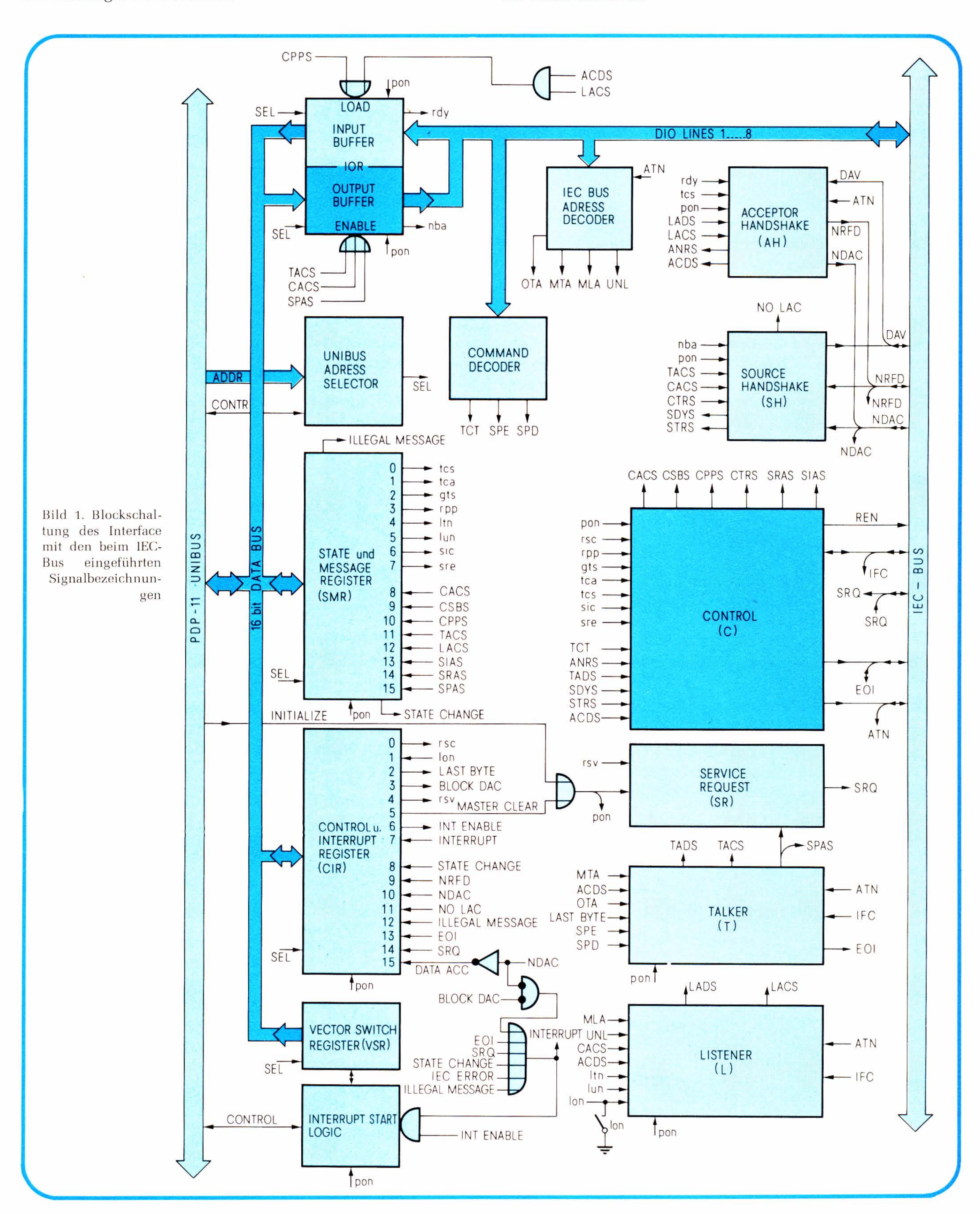
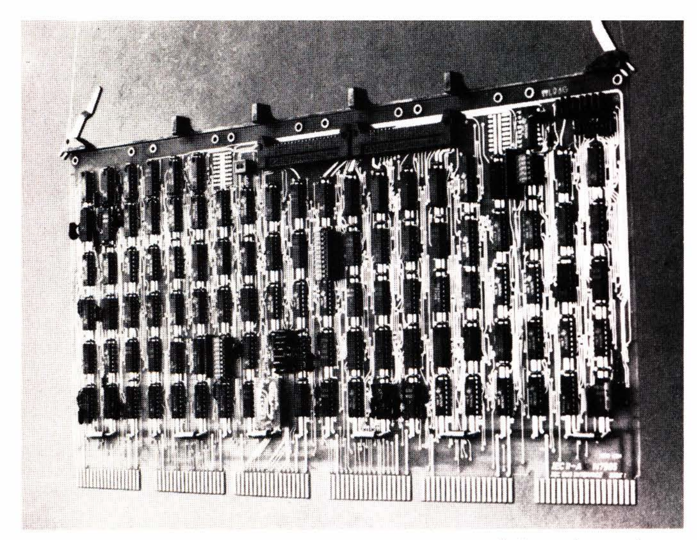


Bild 1. Blockschaltung des Interface mit den beim IEC-Bus eingeführten Signalbezeichnungen



Auf einer Leiterplatte mit den Abmessungen 20 x 39 cm², die in den Rechner PDP-11 eingeschoben werden kann, ist das gesamte IEC-Bus-Interface „IEC11-A" aufgebaut (DEC)

3.1 IEC-Bus-Adreßdecodierer

Da jedes Gerät eine frei wählbare Talker- bzw. Listener-Adresse aufweisen sollte, kann diese im Interface mittels Schalter eingestellt werden. Die gewählte Adresse wird mit den empfangenen Adressen verglichen und dabei verschiedene Signale erzeugt, die dazu dienen, den Talker bzw. Listener zu aktivieren oder zu sperren.

3.2 Talker

Wenn der Talker MTA (*My Talk Address*) vom Adreßdecodierer erhält, ist er adressiert (TADS) und nach Verschwinden des ATN-Signals (*Attention*) aktiviert. Der Aktiv-Zustand wird durch TACS angezeigt, was einerseits den Ausgangspuffer auf die DIO-Leitungen schaltet und andererseits die SH-Funktion (*Source Handshake*) freigibt. Wird nun vom Unibus her ein Datenbyte in den Ausgabepuffer geladen, so wird dies durch nba (*new byte available*) angezeigt. Letzteres veranlaßt die SH-Funktion, das Signal DAV (*Data* Valid) am IEC-Bus zu aktivieren, vorausgesetzt, daß NRFD (*Not ready for data*) nicht aktiv ist, also alle adressierten Listener aufnahmebereit sind. Die Listener quittieren die Übernahme des Datenbyte, indem sie das Signal NDAC (*Not data accepted*) inaktiv machen. Da NDAC auch in die Interrupt-Logik geführt ist, kann dadurch eine Programmunterbrechung im Rechner erzeugt werden, die diesem mitteilt, daß ein neues Datenbyte in den Ausgabepuffer geladen werden kann.
Sobald der Talker vom Adreßdecodierer OTA (*Other Talk Address*) empfängt, wird er inaktiv und kann keine Daten mehr senden. Bevor der Talker jedoch abgeschaltet wird, muß den angeschlossenen Listenern „Übertragungsende" mitgeteilt werden. Dies geschieht dadurch, daß vor dem Laden des letzten Byte in den Ausgabepuffer das Bit „LAST BYTE" im CIR gesetzt wird, was die Aktivierung des EOI-Signals (*End or Identify*) auf dem IEC-Bus zur Folge hat. Eine andere Möglichkeit ist, vor Abschalten des Talkers eine Anzahl sogenannter EOS-Zeichen (*End of String*) zu übertragen, die per Programm definiert werden können. *Bild 2* zeigt den Zeitablauf der Talker-Funktion.

3.3 Listener

Ähnlich wie der Talker wird der Listener durch Empfang von MLA (*My Listener Address*) adressiert. Der Schalter lon (*listen only*) ermöglicht eine Daueradressierung, so daß der Listener laufend mithört. Die „Local Messages" ltn (*local listen*) und lun (*local unlisten*) erlauben das Ein- und Ausschalten des Listener auf internem Wege, ohne eine Listener-Adresse empfangen zu müssen. Der Aktiv-Zustand des Listener wird durch LACS angezeigt, wodurch die AH-Funktion (*Acceptor Handshake*) freigegeben wird.
Der Eingabepuffer sendet rdy (*ready*), wenn er aufnahmebereit ist, was wiederum das IEC-Bus-Signal NRFD inaktiv macht. Wenn die AH-Funktion ein Byte vom IEC-Bus empfängt, wird ACDS (*Accept Data State*) aktiv. Letzteres Signal wird als Übernahmetakt für den Eingangspuffer verwendet. Auch hier wird durch das Ausgehen von NDAC ein Interrupt erzeugt, der dem Rechner mitteilt, daß er das Datenbyte aus dem Eingangspuffer abholen kann. Ist der Puffer geleert, wird rdy erneut aktiviert und der Acceptor kann ein neues Byte übernehmen.
Nach Empfang des Signals UNL (*Unlisten*) wird der Listener inaktiv, wodurch der Empfang weiterer Daten unterbunden wird.

3.4 Control (Steuerfunktion)

Die Control-Funktion (C) erlaubt dem Interface die Verwendung der ATN-Leitung (*Attention*) am IEC-Bus. ATN teilt den angeschlossenen Geräten mit, ob es sich bei dem momentan anstehenden Byte um ein Daten-Zeichen (ATN inaktiv) oder um einen Befehl (ATN aktiv) handelt. Um auf die C-Funktion näher eingehen zu können, ist in *Bild 4* das vereinfachte Zustandsdiagramm wiedergegeben.
Von den abgebildeten Möglichkeiten brauchen natürlich nicht sämtliche implementiert zu sein. Das vorliegende Interface deckt allerdings alle Funktionen ab. Die Normung sieht eine Unterscheidung zwischen einem normalen Controller und einem System-Controller vor. Es können zwar mehrere Geräte mit Control-Funktion angeschlossen sein, jedoch kann nur jeweils *ein* Gerät System-Controller sein. Nur dem System-Controller ist es erlaubt, die Leitungen

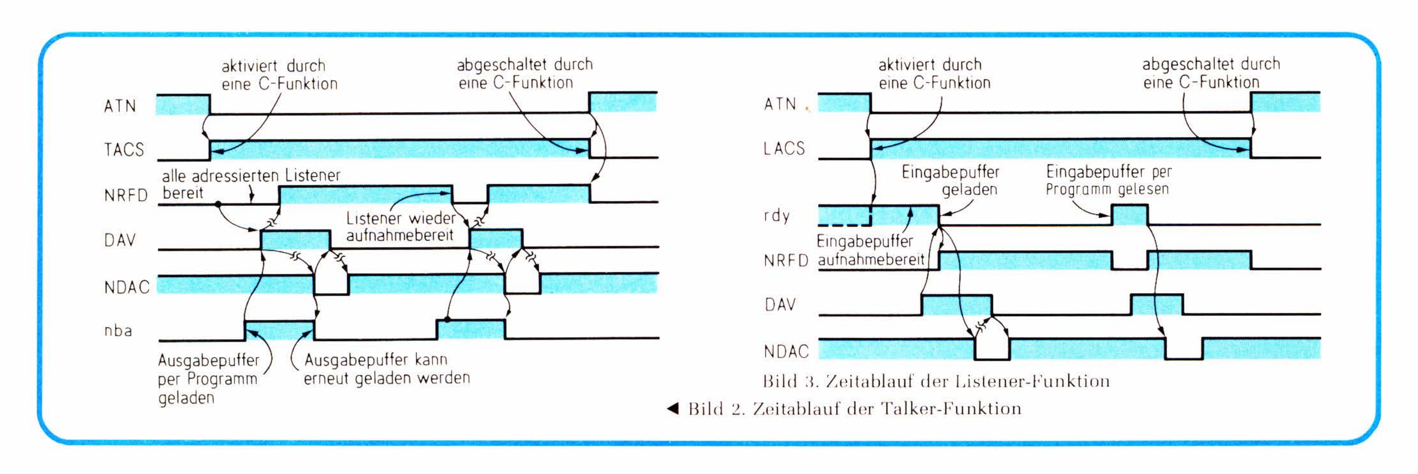


Bild 3. Zeitablauf der Listener-Funktion

◄ Bild 2. Zeitablauf der Talker-Funktion

REN (*Remote Enable*) und IFC (*Interface Clear*) zu aktivieren.
Die „Local Message" rsc (*request system control*) wird durch ein bit im CIR gebildet und kann somit den Zustand SACS (*System Control Active*) per Programm erzeugen.
Werden z. B. vom Programm die bits sic (*send interface clear*) und sre (*send remote enable*) gesetzt, so können sie nur dann SIAS (IFC aktiv) und SRAS (REN aktiv) erzeugen, wenn SACS aktiv ist. Letzterer Zustand ist unabhängig davon, ob die C-Funktion aktiv ist (CACS) oder nicht. Der System-Controller kann also auch den gesamten IEC-Bus in Grundstellung bringen (IFC), wenn die C-Funktion eines anderen Gerätes sich im CACS-Zustand befindet.
Die C-Funktion als solche kann sich grundsätzlich in einem von den folgenden Hauptzuständen befinden:

CACS (*Control Active*)
CSBS (*Control Standby*)
CIDS (*Control Idle*)
CPPS (*Control Parallel Poll*)

Im CACS-Zustand sendet der Controller ATN über den IEC-Bus und ist in der Lage, mittels Source-Handshake Befehle auszugeben, die zuvor vom Programm in den Ausgabepuffer geladen werden müssen.
CSBS ist ein Wartezustand, in dem der Controller bereit ist, durch den vom Rechner gesandten Befehl tcs (*take control synchronously*) oder tca (*take control asynchronously*) wieder aktiviert zu werden. Da naturgemäß die Ausgabe von tcs oder tca asynchron zum IEC-Bus erfolgt, kann bei Verwendung von tca ein gerade in der Übertragung befindliches Datenbyte zerstört werden. Wo dies nicht akzeptabel ist, muß tcs verwendet werden, das nur in Verbindung mit ANRS (*Acceptor not Ready*) wirksam sein kann. Letzteres sorgt dafür, daß der Controller, und damit ATN, zeitlich nur zwischen den Handshake-Zyklen aktiv wird, setzt aber voraus, daß der Listener ständig mithört. Um trotzdem keine Daten zum Rechner transferieren zu müssen, kann BLOCK DAC im CIR gesetzt werden, wodurch dem Acceptor „rdy" gemeldet wird, ohne daß der Eingabepuffer gelesen werden muß. Außerdem werden NDAC-Interrups unterbunden.
CIDS ist der Ruhezustand der C-Funktion. Er wird z. B. dann benötigt, wenn das Interface die Kontrolle über den IEC-Bus einem anderen Gerät zuweisen will. Dazu muß im CACS-Zustand die Talker-Adresse des gewünschten Gerätes gesendet werden und anschließend die Ausgabe des Befehls TCT (*Take Control*) erfolgen. Die C-Funktion des Interface wird wieder aktiv, wenn der momentane Controller dieses als Talker adressiert (TADS) und danach TCT sendet. Mit Erzeugung von SIAS (*sic*) kann die Kontrolle in jedem Falle zurückgeholt werden, unabhängig vom jeweiligen Controller.
Der CPPS-Zustand (*Parallel Poll*) erlaubt den angeschlossenen Geräten die Übergabe eines Statusbyte. Der Controller sendet dazu ATN und EOI aus und übernimmt das Statusbyte in den Eingabepuffer.

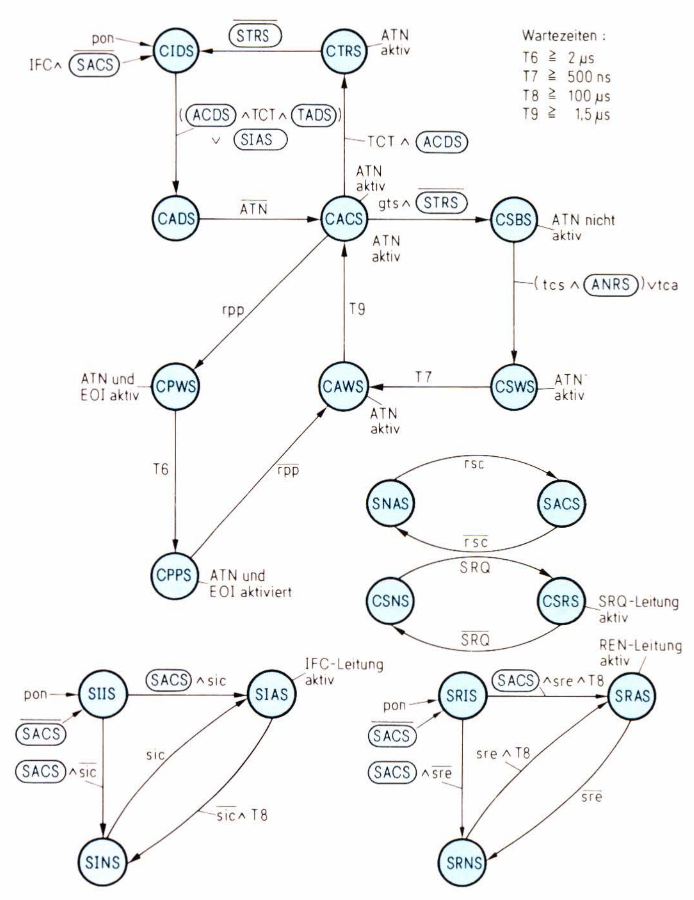


Bild 4. Vereinfachtes Zustandsdiagramm der Control-Funktion „C"

4 Programmierung

In *Bild 5* sind die verwendeten Unibus-Register dargestellt. Die „Local Messages" im SMR können nur gesetzt werden, wenn der zugehörige Status aktiv ist. Z. B. kann tca nur wirksam sein, wenn CSBS aktiv ist. Die Meldung ILL MSGE (*Illegal Message*) im CIR erscheint, wenn ein illegales Bit gesetzt wurde. Irgendeine Änderung der im SMR angezeigten Interface-Zustände führt zur Erzeugung der Meldung STATE CHGE (*State Change*) im CIR.
Die INTERRUPT-Meldung im CIR ist eine ODER-Verknüpfung der Bits im linken Byte mit Ausnahme von Bit 9 und 10. Sie kann einen Interrupt nur dann hervorrufen, wenn INT ENB gesetzt ist. Nachdem eine „Local Message" gesetzt wurde, muß die INTERRUPT-Flagge abgewartet werden, bevor eine neue „Message" erzeugt werden darf.

4.1 *Befehlsausgabe*

Befehle können nur ausgegeben werden, wenn CACS aktiv ist. Das Befehlsbyte wird in den Ausgabepuffer geladen und mittels SH-Funktion übertragen. DATA ACC im CIR zeigt

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
DATA ACC	SRQ	EOI	ILL MSGE	NO LAC	NDAC LINE	NRFD LINE	STATE CHGE	INTER-RUPT	INT ENABLE	MASTER CLEAR	rsv	BLOCK DAC	LAST BYTE	lon	rsc	CIR
SPAS	SRAS	SIAS	LACS	TACS	CPPS	CSBS	CACS	sre	sic	lun	ltn	rpp	gts	tca	tcs	SMR
IN 8	IN 7	IN 6	IN 5	IN 4	IN 3	IN 2	IN 1	OUT 8	OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	IOR
				VECT 11	VECT 10	VECT 9	VECT 8	VECT 7	VECT 6	VECT 5	VECT 4	VECT 3	VECT 2			VSR

Bild 5. Belegung der im Rechner eingebauten Unibus-Register

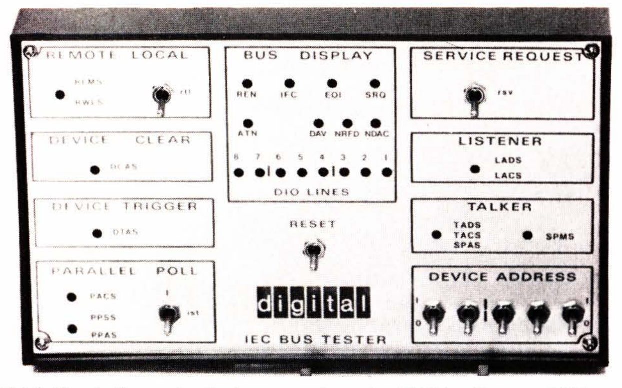


Bild 6. Beschriftung der Bedienungsplatte des IEC-Bus-Testers (DEC)

an, daß der Befehl akzeptiert wurde und ein neues Byte geladen werden darf. Der CACS-Zustand wird verlassen durch Setzen von gts (*go to stand-by*) oder durch Ausgabe des Befehls TCT an einen anderen Controller.

4.2 Senden von Daten

Der Talker kann nur durch Empfang seiner Adresse über den IEC-Bus adressiert werden. Wenn er aktiv wird (TACS), werden die Daten durch Laden des Ausgabepuffers zum IEC-Bus gesendet. Durch DATA ACC wird wiederum angezeigt, daß ein Zeichen von den adressierten Listenern akzeptiert wurde, woraufhin der Puffer neu geladen werden darf. Setzen von LAST BYTE im CIR aktiviert die EOI-Leitung. Wenn das Interface sich im CSBS-Zustand befindet, kann der momentane Talker durch tcs oder tca wieder gesperrt werden (CACS). Dies geschieht normalerweise dann, wenn der Controller EOI empfangen oder das Programm EOS-Zeichen erkannt hat.

4.3 Empfangen von Daten

Der Listener kann sowohl durch Empfang seiner Adresse aktiviert werden als auch durch ltn, wenn CACS aktiv ist. Durch lon in CIR wird der Listener permanent adressiert. Wenn der Listener aktiv ist (LACS), wird durch DATA ACC der Empfang eines Datenbyte angezeigt, so daß der Eingabepuffer gelesen werden kann. Erst nach dem Lesen des Puffers ist die AH-Funktion wieder aufnahmebereit für ein neues Byte.

4.4 Serial Poll

Wenn SRQ im CIR durch ein angeschlossenes Gerät aktiviert wurde, muß zunächst der Befehl SPE (*Serial Poll Enable*) gesendet werden. Danach muß der Controller einen der angeschlossenen Talker adressieren und durch gts in den CSBS-Zustand gebracht werden. Das angesprochene Gerät sendet nun sein Statusbyte zum Eingangspuffer. Diese Sequenz wird so lange mit den übrigen Talker-Adressen wiederholt, bis ein Statusbyte erscheint, in dem Bit 7 gesetzt ist, wodurch das betreffende Gerät als Erzeuger von SRQ identifiziert ist. Danach schaltet das Gerät SRQ automatisch ab. Die übrigen Bits im Statusbyte können für zusätzliche, gerätespezifische Mitteilungen verwendet werden.



Ing. (grad.) Uwe Richert stammt aus Westfalen und studierte an der Staatlichen Ingenieurschule Dortmund Nachrichtentechnik mit Abschluß 1967. Nach kurzem Zwischenspiel bei der Siemens AG in München war er – in der gleichen Stadt verbleibend – zwei Jahre im Ingenieurbüro der Firma Kienzle Apparate mit der Entwicklung von Prüfgeräten beschäftigt, dann drei Jahre bei der PCS GmbH (Entwicklung von Computerperipherie und Datenerfassungssystemen). Seit 1973 arbeitet er bei der Firma Digital Equiqment (DEC) im Projekt-Management und an der Entwicklung von Computer-Spezialsystemen. 1974 wurde er dortselbst „Engineering Unit Manager". Hobbys: Musik, Fotografie
ELEKTRONIK-Leser seit 1966

4.5 Parallel Poll

Parallel Poll benutzt keine Interrupt-Leitung wie SRQ. Vielmehr wird verschiedenen Geräten ein eigenes Bit im Statuswort zugeordnet. Außerdem brauchen sie nicht als Talker adressiert zu werden.

Um die PP-Funktion auszuführen, müssen die gewünschten Geräte zunächst als Listener adressiert werden. Danach sendet der Controller PPC (*Parallel Poll Configure*), wodurch diese in die Lage versetzt werden, am Parallel Poll teilzunehmen. Wenn nun der Controller PPE (*Parallel Poll Enable*) aussendet, sind die Geräte bereit und warten darauf, daß der Controller IDY und ATN aktiviert. Dies geschieht durch Setzen von rpp im SMR, wodurch CPPS aktiv wird. Während dieses Zustandes wird das Statuswort in den Eingabepuffer geladen. Durch Rücksetzen von rpp geht der Controller wieder in seinen CACS-Zustand, in dem er PPD (*Parallel Poll Disable*) senden kann.

4.6 Control-Austausch

Das Interface ist in der Lage, die Steuerung (Control) des IEC-Bus an ein anderes Gerät zu übergeben. Dazu wird letzteres als Talker adressiert und TCT gesendet, worauf CACS im SMR ausgeht und damit die Übergabe anzeigt. Wenn das Interface in diesem Zustand andererseits TCT empfängt, wird CACS erneut aktiv. Setzen von sic im SMR, wenn rsc ebenfalls gesetzt ist, aktiviert in jedem Falle CACS.

5 Testmöglichkeiten

Aufgrund der besonderen Struktur des IEC-Bus kann der Datentransfer des Interface voll ausgetestet werden, ohne irgendein Gerät anschließen zu müssen, da der Listener gleichzeitig empfangen kann, wenn der Talker sendet.

Man kann auch mehrere dieser Interfaces zusammenschalten, wodurch zusätzlich eine sehr interessante Anwendung erreicht wird, nämlich eine Rechnerkopplung zwischen zwei oder mehreren Rechnern (*Interprocessor Link*).

Um jedoch sämtliche Funktionen testen zu können, wurde ein spezielles Testgerät entwickelt, dessen Bedienungsfeld in *Bild 6* dargestellt ist. Das Testgerät kann wie ein normales IEC-Bus-Gerät angeschlossen werden, erhält aber zusätzlich seine Stromversorgung ebenfalls vom Interface.

In der Mitte befindet sich ein Anzeigefeld aus Leuchtdioden, das die 16 IEC-Bus-Leitungen darstellt. Die Geräteadresse ist mit fünf Kippschaltern individuell einstellbar.

Ob der Talker bzw. Listener adressiert ist, wird durch entsprechende Leuchtdioden angezeigt. Zur Aktivierung der SRQ-Leitung kann die Taste „rsv" (*request service*) gedrückt werden. Wenn der momentane Controller daraufhin einen „Serial Poll" ausführt, wird dies durch SPMS (*Serial Poll Mode*) angezeigt.

PACS (*Parallel Poll Addressed to Configure*) zeigt an, daß das Gerät vom Controller zur Teilnahme an einem „Parallel Poll" selektiert wurde, wohingegen PPSS (*Parallel Poll Standby*) meldet, daß das Gerät bereit ist, auf den Empfang von ATN△EOI mit der Übergabe eines Status-Bit zu reagieren. Der Schalter „ist" (*individual status*) ermöglicht das Einstellen des Status-Bit.

Die Indikatoren DTAS (*Device Trigger Active*) und DCAS (*Device Clear Active*) geben ein Blinksignal ab, wenn ein Trigger- bzw. Clear-Befehl empfangen wurde.

Wenn das Gerät vom Controller auf Fernbedienung programmiert ist, leuchtet REMS (Remote) auf. Die Taste „rtl" (*return to local*) ermöglicht, das Gerät manuell in den „Local"-Zustand zu bringen.

Literatur

1 *Klaus, J.*: Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.

Dr. Walter Haupt
Ing. (grad.) Wolfgang Otternberg

Der IEC-Bus für größere Entfernungen

Als vor gut sieben Jahren in einem Bericht in der ELEKTRONIK die Forderung nach einem Bussystem zur Verbindung von Meßgeräten zu Meßsystemen erhoben wurde, zeichneten sich bereits zwei Anwendungsgebiete ab [1]. Priorität sollte damals der Laborbereich haben mit Entfernungen bis etwa 20 m. Darüber hinaus wurde bereits ein industrieller Anwendungsbereich erkannt. Für die erstgenannte Anwendung kann man die Normenarbeit jetzt praktisch als abgeschlossen betrachten. Es ist deshalb an der Zeit, sich dem zweiten Aufgabenbereich zuzuwenden, zumal dies durch eine geringfügige Modifikation der Norm erreichbar ist. Dieser Aufgabe kann man sich nicht entziehen mit einem Hinweis auf vollserielle Schnittstellen, die irgendwann einmal genormt sein werden. Derartige Schnittstellen sind auch dringend erforderlich; sie können aber bei schneller Echtzeitverarbeitung eine so leistungsfähige Schnittstelle wie den IEC-Bus nicht ersetzen.

1 Voraussetzung für eine „Industrieversion" des IEC-Bus

Der IEC-Bus ist konzipiert für „...Schnittstellensysteme zur Verbindung von ...Meßgeräten..., aus denen Meßsysteme zusammengestellt werden". So heißt es unter 1.1.1 im Geltungsbereich von E DIN-IEC 66.22 [2]. Schwerpunkt dieser Betrachtung ist die Labormeßtechnik. In diesem Fall ist auch die Einschränkung unter 1.1.2 des Geltungsbereichs von geringer Bedeutung: „...Schnittstellen in Meßsystemen..., bei denen... die Gesamtlänge der Verbindungskabel, die den Übertragungsweg bilden, 20 m nicht überschreitet".

In großen Laboratorien, Prüffeldern, Technika oder gar bei der betrieblichen Meßdatenerfassung in Meßwarten großer verfahrenstechnischer Anlagen ist diese enge räumliche Begrenzung nicht mehr tragbar. Dies haben die Herausgeber der Norm vorausschauend bereits berücksichtigt und unter 1.1.2 des Geltungsbereichs entsprechend erwähnt: „Die grundlegenden funktionellen Festlegungen der vorliegenden Norm können auch bei Anwendungen von Digital-Schnittstellen verwendet werden, bei denen größere Entfernungen, eine größere Anzahl Geräte, erhöhte Störsicherheit gegenüber der Umgebung, bzw. Kombinationen dieser Bedingungen erforderlich sind. Bei diesen erweiterten Anwendungen können andere elektrische und mechanische Festlegungen zutreffen."

Bisher wurde einem solch erweiterten Anwendungsbereich seitens der Gerätehersteller allerdings keine Beachtung geschenkt, so daß einem interessierten Anwender keine Geräte zur Verfügung standen. So auch der Bayer AG, Leverkusen, einem Anwender aus der chemischen Industrie, der in einer Eigenentwicklung den IEC-Bus für die rechnergestützte betriebliche Meßdatenerfassung und -verarbeitung nutzbar zu machen hatte [3]. Die Zielsetzung bei der Entwicklung dieser busorientierten Prozeßperipherie war:

1. Benutzung eines funktionell ausgereiften und genormten Buskonzepts;
2. Unabhängigkeit vom Rechnerfabrikat;
3. Möglichkeit des Einsatzes funktionell gleicher Geräte in Laboratorien, Technika und Betrieben;
4. Leichte Systemprüfbarkeit über eine einzige Schnittstelle;
5. Eindeutige Abgrenzung der technischen Verantwortungsbereiche.

Die Erfahrungen mit einer größeren Anzahl derartiger Anlagen bestätigen die Richtigkeit der Zielsetzung.

Die Realisierung einer erweiterten Anwendung des IEC-Bus wird begünstigt durch den „modularen" Aufbau der Norm. So sind praktisch rückwirkungsfrei auf die funktionellen Festlegungen lediglich einige Abweichungen von der Norm bei den Treiber- und Empfängerschaltungen, den Kabeln und Steckern notwendig. Dazu müssen auch noch einige wenige Betrachtungen hinsichtlich des Zeitverhaltens angestellt werden, die sich jedoch auf das Steuergerät (Controller) beschränken.

Die Begrenzung des Übertragungsweges ist im wesentlichen gegeben durch die elektrischen Eigenschaften der Leitungstreiber, -empfänger und der Verbindungskabel. Die Treiber- und Empfängerspezifikationen sind durch die TTL-Technologie bestimmt. Das spezifizierte 24adrige Kabel mit verdrillten Aderpaaren für die wichtigsten Leitungen des Steuerbusses, jedoch unverdrillten Leitungen für den Datenbus, ist wegen der unterschiedlichen Signallaufzeiten auf Daten- und Steuerbus wohl nur im extremen Nahbereich sinnvoll verwendbar. Bei größeren Entfernungen ist es ratsam, für alle Signale identische Leitungen zu verwenden, d. h. alle Signale über verdrillte Adernpaare zu übertragen. Sucht man jetzt eine geeignete und genormte Technik zur Datenübertragung, so bietet sich die symmetrische Doppelstrom-Schnittstelle

nach CCITT[1]) V 11 bzw. X 27 an [4]. Sie stimmt überein mit dem EIA[2])-Standard RS-422 [5].

2 Realisierungsvorschlag für eine „Industrieversion"

2.1 Pegel und Treiberstufen

Eine Industrieversion des IEC-Bus sollte für eine normale Ausdehnung von etwa 100 m konzipiert sein, jedoch Bedingungen beinhalten, die eine Ausdehnung bis etwa 1000 m erlauben.

Man ersetzt die Leitungstreiber und -empfänger des IEC-Bus durch Differenztreiber und -empfänger entsprechend der Spezifikation CCITT V 11 (*Bild 1*). Dazu werden die jeweiligen Signalleitungen definiert als A- und B-Leitung mit entsprechend festgelegten Signalpegeln:

Logischer Zustand	Elektrischer Signalpegel A-Leitung	B-Leitung
0	hoch	niedrig
1	niedrig	hoch

Der Pegel auf der A-Leitung entspricht der ursprünglichen Norm.

Die handelsüblichen Differenztreiber mit den nach V 11 festgelegten Spezifikationen sind durchweg als Tristate-Treiber erhältlich. Für die Signale NRFD, NDAC und SRQ sowie die bei Parallelabfrage PP ausgewählte Datenleitung ist eine „Wired-Or-Verknüpfung" erforderlich. Sie ist möglich durch eine Verknüpfung des Signals mit der Ausgangssteuerung (output control). *Bild 2* zeigt Realisierungsmöglichkeiten. Bei Treibern, die über getrennte Ausgänge nach 0 bzw. U_{cc} schalten, kann man die A-Leitung über den offenen Kollektor-Ausgang, die B-Leitung über den offenen Emitter-Ausgang treiben, siehe auch [6].

[1]) Comité Consultatif International Télégraphique et Téléphonique – CCITT
[2]) Electronic Industries Association

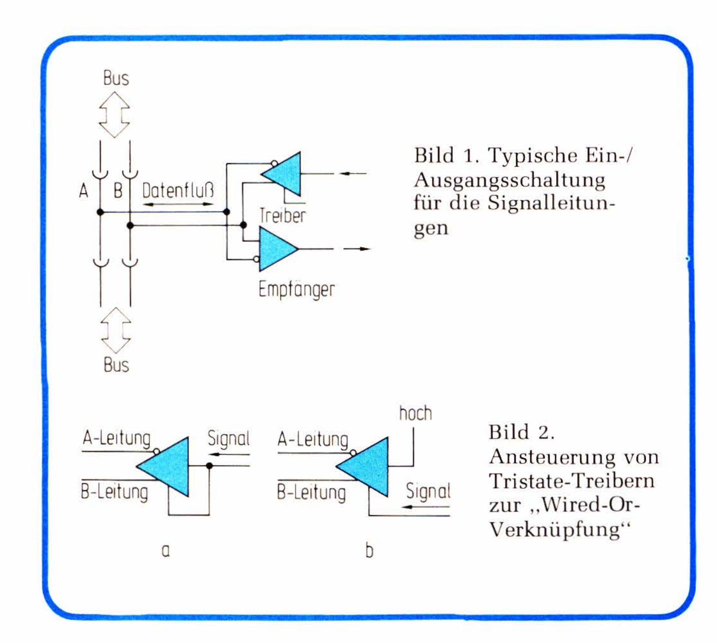


Bild 1. Typische Ein-/Ausgangsschaltung für die Signalleitungen

Bild 2. Ansteuerung von Tristate-Treibern zur „Wired-Or-Verknüpfung"

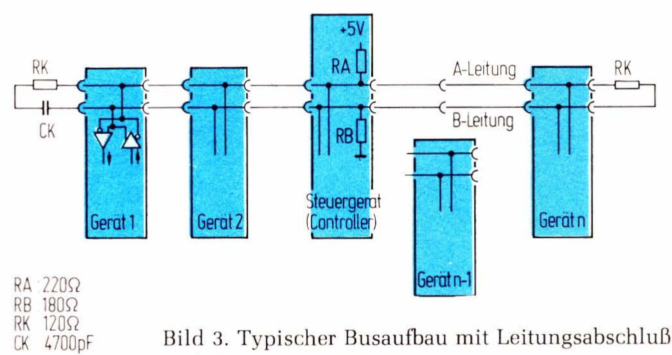


Bild 3. Typischer Busaufbau mit Leitungsabschluß und Versorgungserzeugung

DIN-IEC 66.22 fordert, daß passiv „logisch 0" gesendet werden soll. Dies ist durch entsprechende Vorspannung der A- und B-Leitung erreichbar. Anstatt jedes Gerät mit (fehlangepaßten) Eingangswiderständen zu versehen, empfiehlt es sich, einen ordentlichen Leitungsabschluß zu wählen und die Vorspannung dann durch Widerstände lediglich im Steuergerät zu erzeugen.

Die Widerstände R_A und R_B sind nur im Steuergerät erforderlich. Sie müssen bei einer Übertragung der Steuerfunktion jeweils im alten Steuergerät ab- und im neuen Steuergerät zugeschaltet werden. Einer der jeweiligen Widerstände R_A oder R_B kann durch eine Konstantstromquelle von ca. 10 mA ersetzt werden (*Bild 3*).

2.2 Leitungen und Anschlüsse

Der Kabelabschluß erfolgt einseitig direkt; auf der anderen Seite ist der Abschlußwiderstand kapazitiv angekoppelt. Infolge des geräteunabhängigen Kabelabschlusses ist das Zeitverhalten und damit die maximale Übertragungsrate unabhängig von der Anzahl der angeschlossenen Geräte. Durch den Verzicht auf eine Widerstandsbeschaltung in den einzelnen Geräten ist es im Gegensatz zum Standard-IEC-Bus auch unkritisch, wieviele Geräte am Bus ausgeschaltet sind.

Das Verbindungskabel soll mindestens 16 verdrillte Adernpaare, eine Leitung für die Digital-Masse haben und obendrein geschirmt sein. Dazu eignen sich die üblichen 18- oder 20paarigen Computer- oder Fernmeldekabel. Die nicht als Datenleitungen benötigten Adernpaare sind als Masseleitung zu verwenden.

Als Stecker sollte man auf einen gebräuchlichen und möglichst genormten Stecker mit mindestens 35 Kontakten zurückgreifen. 32 Kontakte werden für die Datenleitungen benötigt. Mindestens einer soll für die Digital-Masse zur Verfügung stehen; der äußere Schirm sollte auf mindestens zwei Kontakte aufgelegt werden. (Der oben erwähnte Anwender aus der chemischen Industrie hat der rauhen Einsatzbedingungen wegen den 39poligen Steckverbinder nach DIN 41618 und IEC 130-6 gewählt.) Es empfiehlt sich, statt Verwendung eines „Huckepack-Steckers" die Leitungen durch die Geräte zu schleifen und diese mit je einer Stecker- und Buchsenleiste zu versehen. Beim Abtrennen eines Gerätes vom Bus kann dann das Buskabel durchverbunden werden, ohne daß Steckerstifte offen bleiben; siehe auch Bild 3, Gerät n-1. Die Kabelabschlußwiderstände können beidseitig in Ab-

Tabelle. Vorgeschlagene Zeitangaben

Zeiten	Funktion	Beschreibung	Wert bis 100 m	Wert je weitere 100 m
T_1	SH	Einschwingzeit für Mehrdrahtnachrichten	$\geqq$700 ns*)	–
T_2	SH, AH, T, L	Antwort auf ATN	$\leqq$200 ns	–
T_3	SH	Schnittstellennachricht-Empfangszeit	>0	–
T_4	T, TE, L, LE, C	Antwort auf IFC oder REN falsch	<100 µs	–
T_5	PP	Antwort auf ATN ∧ EOI	$\leqq$200 ns	–
T_6	C	Parallelabfrage-Ausführungszeit	$\geqq$ 2 µs	1 µs
T_7	C	Verzögerung der Steuereinheit, um dem augenblicklichen Sprecher das Erkennen der ATN-Nachricht zu ermöglichen	$\geqq$ 1 µs	1 µs
T_8	C	Dauer von IFC oder REN falsch	>100 µs	–
T_9	C	Verzögerung für EOI	$\geqq$ 1 µs	1 µs

*) T_1 kann 500 ns für alle weiteren Bytes betragen, die auf das erste Byte nach dem Übergang ATN falsch folgen.

schlußstecker gelegt werden, beispielsweise der ohmsche ans buchsenseitige Kabelende, der kapazitiv angekoppelte Abschluß an das steckerseitige Ende.

2.3 Zeitbedingungen und Übertragungsraten

Die Zeitangaben für die Zustandsübergabe bleiben gegenüber der ursprünglichen Norm unverändert für alle Geräte, die keine Steuerfunktion ausüben (T_1...T_5). Für T_1 gilt die Aussage für Tristate-Treiber. Bei Entfernungen bis 100 m bleiben die Werte für T_6 und T_8 unverändert. T_7 und T_9 werden auf 1 µs erhöht. Bei Übertragungswegen über 100 m müssen die Zeiten T_6, T_7 und T_9 jeweils um 1 µs/100 m erhöht werden. Die *Tabelle* zeigt die jeweiligen Zeitangaben.

Sollten bei großer Ausdehnung des gesamten Übertragungswegs zwischen zwei nahe benachbarten Geräten hohe Datenraten schnell übertragen werden, so ist dies möglich, sofern der Hörer als einziger Hörer adressiert ist. Dies ist durch eine spezielle Adressierung oder einen ACG-Befehl realisierbar. In diesem Fall kann der alleinige Hörer die Nachrichten $\overline{\text{NRFD}}$ und $\overline{\text{NDAC}}$ aktiv senden, also „hart“ quittieren. In diesem Fall entfällt für den Hörer die Verknüpfung der Signale NRFD und NDAC mit der Ausgangssteuerung nach Bild 2. Wie Versuche zeigten, ist es dann möglich, bei einer Gesamtlänge des Übertragungswegs von 100 m und mehr zwischen benachbarten Geräten, die 10 m voneinander entfernt sind, Datenraten von über 1 MByte/s zu übertragen (siehe auch [6]).

3 Ausblick

Anfang 1975 zeichnete sich in der Normung des IEC-Bus bereits ein „Sieg der Vernunft“ ab, wie der ELEKTRONIK-Leitartikel in Heft 1/1975 zu recht behauptete [7]. Sichtbar wurde dieser erste „Sieg“ Ende 1975, als der Normenentwurf unter 66(CO)22 seitens der IEC und dann im Januar 1976 bereits die deutsche Übersetzung als Entwurf DIN-IEC 66.22 veröffentlicht wurden. Die endgültige Norm ist funktionell mit den Normenentwürfen identisch. Nicht zu den „Siegern“ können sich die Betriebsmeßtechniker rechnen, denen der IEC-Bus zu „klein“ ist. Auch sie wollen programmierbare elektronische Meßgeräte einsetzen und miteinander zu Systemen verbinden. Das verpflichtet die Normer des IEC-Bus auch weiterhin zur Vernunft und zur Zusammenarbeit, zumal in keiner Nachbardisziplin in absehbarer Zeit ein vergleichbares weltweit genormtes System zu erwarten ist, das man statt des IEC-Bus im industriellen Bereich einsetzen kann.

Literatur

[1] *Schmidt, U.:* Genormte BUS-Systeme sind eine zwingende Notwendigkeit. ELEKTRONIK 1973, H. 8, S. 300.

[2] DIN-IEC 66.22 Elektrische Meßtechnik. Byteserielles, bitparalleles Schnittstellensystem für programmierbare Meßgeräte. Beuth-Verlag GmbH, Berlin, Köln.

[3] *Klein, P. E., Wilhelmy, H. J.:* Der IEC-BUS, Stand der Normung und Entwicklung – Marktübersicht – Anwendungen. ELEKTRONIK 1977, H. 10, S. 63...74.

[4] CCITT-Empfehlungen der V-Serie und der X-Serie: Datenübertragung. 3. Auflage 1977. R. v. Decker's Verlag, G. Schenk, Heidelberg, Hamburg.

[5] EIA-STANDARD RS-422. Electrical Characteristics of Balanced Voltage Digital Interface Circuits. Electronic Industries Association, Engineering Department, Washington.

[6] *Ehnert, D.:* Das Zeitverhalten einer IEC-Bus-Version für größere Entfernungen. ELEKTRONIK 1978, H. 4, S. 62...64.

[7] *Freytag, H. H.:* Ein Sieg der Vernunft (Leitartikel). ELEKTRONIK 1975, H. 1, S. 3.

[8] *Klaus, J.:* Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.

Ing. (grad.) Dietmar Ehnert

Das Zeitverhalten einer IEC-Bus-Version für größere Entfernungen

Wie die Verfasser des DIN-Entwurfes selber schreiben [3], kann die auf zwanzig Meter beschränkte Form des IEC-Busses nur in Laboratorien und Prüffeldern Verwendung finden. Damit ist sie für die Industrie indiskutabel, denn dort müssen Entfernungen um einhundert Meter oder mehr überbrückt werden, z. B. bei der Meßwerterfassung für Prozeßrechner. Dafür kommen u. a. zwei Lösungen in Frage:

- Datenübertragungssysteme mit erhöhten Sendepegeln
- Datenübertragungssysteme mit Differenzsignalen.

Um die Eigenschaften verdrillter Adernpaare nutzen zu können, entschied man sich bei einer Entwicklung der Bayer AG in Leverkusen (vgl. den Beitrag in diesem Heft: Der IEC-Bus für größere Entfernungen) für das symmetrisch aufgebaute Übertragungssystem, das mit Differenzsignalen arbeitet. Dabei werden Störungen, die galvanisch, induktiv oder kapazitiv auf ein Adernpaar gekoppelt werden, vom Differenzempfänger gleichtaktunterdrückt. So kann mit einem größeren Materialaufwand (doppelte Anzahl an Leitungstreibern und Adern) eine sehr geringe Störempfindlichkeit erreicht werden.

Als Kabel wird ein „Computerkabel" nach DIN 47 414 mit Kupfergeflecht-Abschirmung und zwanzig Adernpaaren verwendet, die aus feindrähtiger Kupferlitze von 0,14 mm² bestehen und einen Wellenwiderstand von etwa 100 Ω besitzen. Die einzelnen Adern wie auch das Gesamtkabel (D = 15,5 mm) sind mit PVC isoliert und geschützt.

1 Schaltungstechnik

Zur Erläuterung der Schaltungstechnik sei daran erinnert, daß die Steuerbefehle ATN, IFC und REN auf dem IEC-Bus ausschließlich vom Steuergerät gesendet werden. So ist es möglich, diese Signale mit Gegentakt-Ausgängen auf die Leitung zu schicken, wie es in *Bild 1* im ersten Beispiel (Eingang 1) gezeigt ist. Dabei liegt die Freigabe auf H-Pegel.

Die Nachrichten auf den Leitungen DIO, DAV und EOI können sowohl von dem Steuergerät als auch von den Sprechern gesendet werden. Auch hier beschreibt der Normentwurf offene Kollektor-Ausgänge, läßt aber auch Tristate-Ausgänge zu. Dabei wird die gleiche Anordnung wie im ersten Beispiel verwendet. Es muß nur darauf geachtet werden, daß immer nur ein einzelnes Gerät, sei es Steuergerät oder Sprecher, zum gleichen Zeitpunkt senden kann.

Die Signale NRFD, NDAC und SRQ müssen Wired-OR-verknüpft sein, um von mehreren Geräten gleichzeitig auf die Leitung gesendet werden zu können. Diese Verknüpfungsart läßt sich durch zwei Schaltungsarten des Tristate-Treibers realisieren. Sie sind im zweiten (Eingang 2) und dritten Beispiel (Eingang 3) gezeichnet.

Im zweiten Beispiel fehlt die Brücke zwischen dem offenen Emitter (tritt oberhalb der Spitze aus) und offenen Kollektor, der aus der Spitze des Verstärkers austritt. Diese Beschaltung stellt für die A-Leitung einen offenen Kollektor-Ausgang und für die B-Leitung

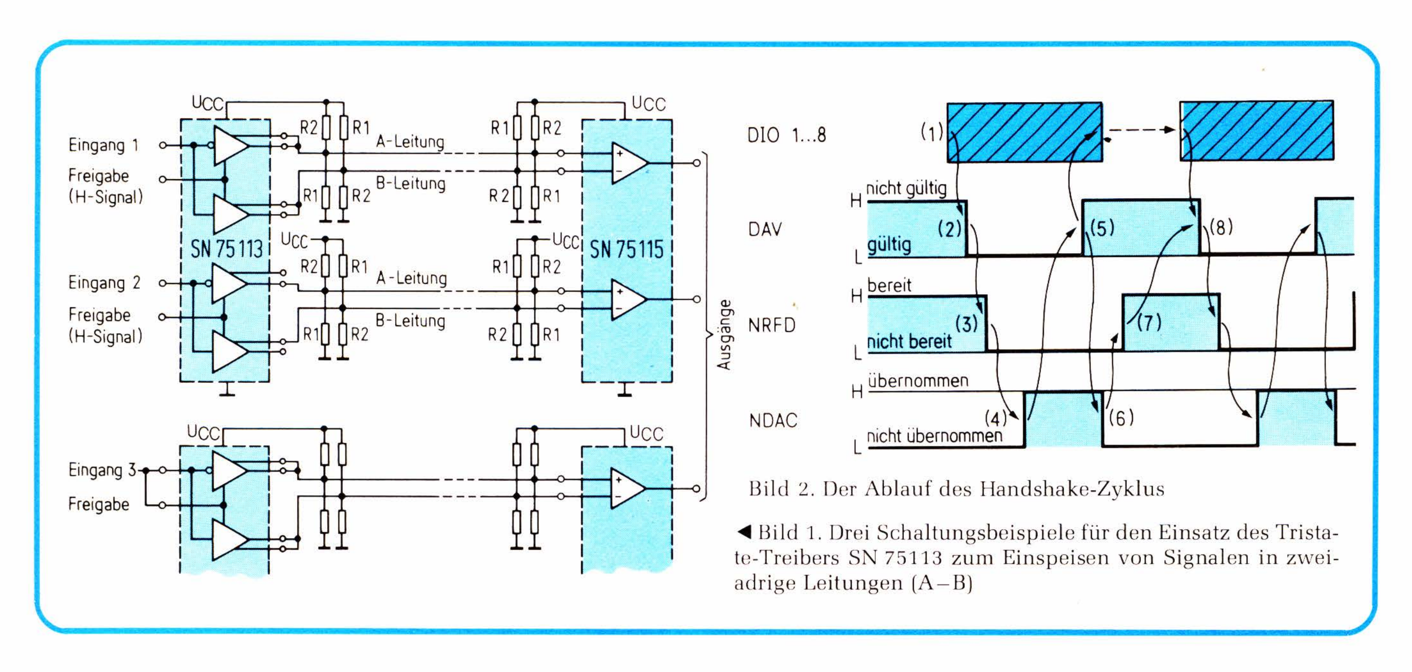


Bild 2. Der Ablauf des Handshake-Zyklus

◀ Bild 1. Drei Schaltungsbeispiele für den Einsatz des Tristate-Treibers SN 75113 zum Einspeisen von Signalen in zweiadrige Leitungen (A–B)

Wahrheitstabelle des Leitungstreibers SN 75 113

Freigabe	Eingang	A-Leitung	B-Leitung
L	X	hochohmig	hochohmig
H	L	High	Low
H	H	Low	High

einen offenen Emitter-Ausgang dar, so daß mehrere Sender gleichzeitig die Leitung benutzen können.

Im dritten Beispiel ist die Freigabe mit dem Eingang zusammengeschaltet. Liegt nun ein H-Signal am Eingang an, schaltet der Tristate-Treiber die Leitung hart gegen GND. Ein L-Pegel am Eingang bzw. an der Freigabe gestaltet den Ausgang hochohmig. Das Verhalten des verwendeten Leitungstreibers zeigt seine *Wahrheitstabelle*.

2 Meßverfahren

Sollen Daten von einem Gerät an einen oder mehrere Hörer gesendet werden, läuft der Transport nach einem gewissen Schema ab. Dieser *Handshake-Prozeß* [2] bestimmt die maximale Übertragungsrate (*Bild 2*). Es wurde eine Schaltung entwickelt, die diesen Zyklus steuert und automatisch ablaufen läßt. Es wurden zwei Stationen – ein Steuergerät und ein Communicator [1]) – mit einer definierten Kabellänge verbunden.

Bei einer Zyklusfrequenz von 10 kHz wurde die Laufzeit (in einer Richtung) mit einem Zweistrahl-Oszillografen bestimmt. Um praxisnahe Aussagen zu erhalten, wurden die Logiksignale vor dem Bus-Treiber-Eingang und beim zweiten Gerät nach dem Bus-Empfänger-Ausgang dargestellt; dann kann man die Zeitdifferenz zwischen beiden Signalen auf dem Bildschirm ablesen.

[1]) Communicator = Gerät, das die Logik des Übergabebusses steuert und sich als Hörer oder Sprecher adressieren läßt, vgl. Bild 5.

Bild 3 zeigt die gemessenen Zeiten der Wired-OR-verknüpften Signale NRFD und NDAC und die des Gegentakt-Ausganges DAV. Um den Betrieb von zehn angeschlossenen Stationen zu simulieren, wurde anstatt der Standardabschlußplatine, die mit einem Spannungsteiler von R 1 = 6800 Ω und R 2 = 3300 Ω je Leitung bestückt ist, eine Widerstandskombination von R 1 = 680 Ω und R 2 = 270 Ω verwendet.

Das Gegentakt-Signal DAV besitzt auf der Leitung steile Anstiegs- und Abfallflanken. Die Auswertung der Meßergebnisse ergab:

- Es ist gleichgültig, mit welchem Widerstand das Adernpaar abgeschlossen ist.
- Entscheidend für die Laufzeit der Signale ist die Distanz vom ersten bis zum letzten Gerät. Die Informationen werden fast mit 1/2 Lichtgeschwindigkeit übertragen.

Die Signale NRFD und NDAC, die Wired-OR-verknüpft sind, besitzen steil ansteigende Flanken (Schalten der A-Leitung gegen Masse und der B-Leitung gegen Versorgungsspannung), während im entgegengesetzten Fall die Leitungskapazitäten über die Widerstände R 2 entladen werden müssen. Hier ergab die Auswertung folgende Ergebnisse:

- Die Anzahl der angeschlossenen Geräte bzw. der Abschlußwiderstand des Adernpaares bestimmen die Länge der Laufzeit.
- Die gesamte angeschlossene Kabellänge entscheidet über die Übertragungsgeschwindigkeit.

Das Durchhängen der Kurven läßt sich auf bausteininterne Verzögerungszeiten zurückführen.

3 Untersuchung des Handshake-Zyklus

Für den Anwender des IEC-Busses ist es entscheidend zu wissen, wieviele Datenbyte pro Zeiteinheit von diesem System in Abhängigkeit von der Kabellänge übertragen werden können. Die Frequenz wurde bis zu dem Punkt erhöht, an dem sich die Zykluszeit auf dem Oszillografenschirm nicht mehr verringerte.

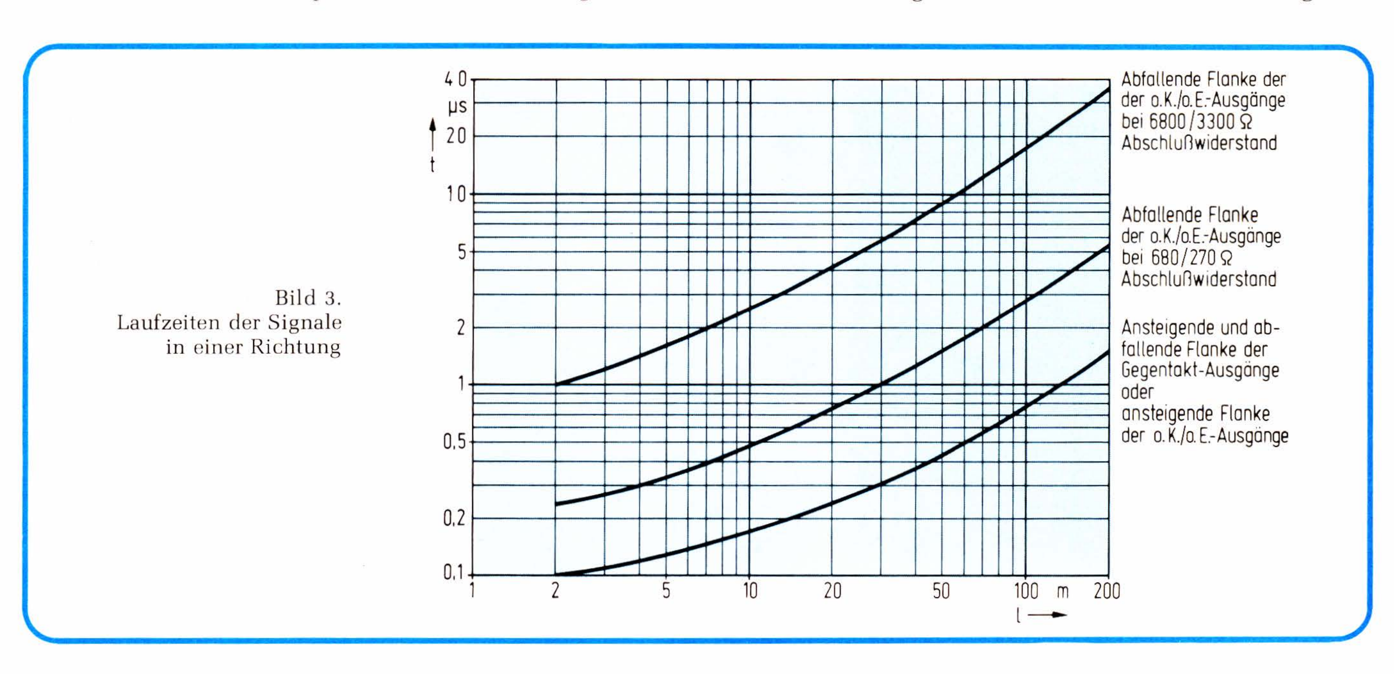


Bild 3. Laufzeiten der Signale in einer Richtung

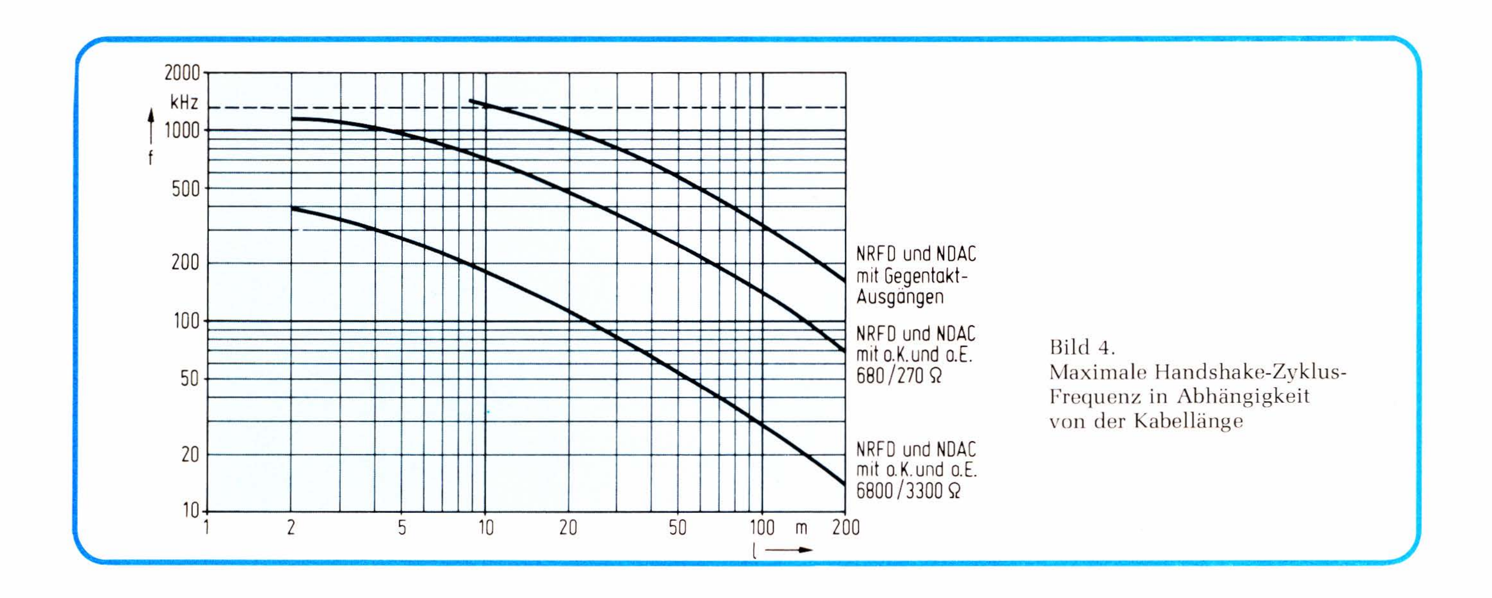


Bild 4. Maximale Handshake-Zyklus-Frequenz in Abhängigkeit von der Kabellänge

Damit war der schnellstmögliche Handshake-Zyklus erreicht (*Bild 4*).

Die maximale Zyklusfrequenz des IEC-Busses wird sich zwischen den beiden Kurven befinden, die Wired-OR-verknüpft sind. Bei diesen Kurven ist die Frequenz nicht von dem maximalen Abstand (d. h. vom Abstand zwischen erstem Gerät und letztem Gerät auf der Leitung) abhängig, sondern, wie die Messungen der Laufzeit in einer Richtung bestätigen, von der angeschlossenen Kabellänge und den Abschlußwiderständen. Ein Handshake-Zyklus setzt sich zusammen aus:

1. Ausgleichsvorgang des Signals NRFD,
2. Laufzeit des Signals DAV,
3. Ausgleichsvorgang des Signals NDAC,
4. Laufzeit des Signals DAV und
5. interne Verzögerungszeit der Busanschlußlogik.

4 Ergebnisse

Die modifizierte Version des IEC-Busses erlaubt es dem Anwender, den Bus auch über längere Distanzen einzusetzen. Weil Differenzsignale auf die Leitung gesendet werden, nutzt man die günstigen Eigenschaften der verdrillten Adernpaare gut aus. Deswegen konnte z. B. noch eine maximale Datenflußrate von 140 kByte/s über 100 m Kabel realisiert werden, als zehn angeschlossene Geräte simuliert wurden (*Bild 5*).

Wird nun der Abschlußwiderstand dem Wellenwiderstand der Leitung von etwa 100 Ω angepaßt (rein ohm'scher Abschluß), ergeben sich daraus zwei Vorteile: Erstens wird die Reflexion auf der Leitung vermieden, und zweitens entlädt sich die Leitungskapazität schneller, so daß eine noch höhere Übertragungsrate mit den Wired-OR-verknüpften Signalen NRFD und NDAC erzielt werden kann. Die Abschlußwiderstände werden zweckmäßig am ersten und letzten Gerät angeschlossen.

Ferner besteht die Möglichkeit, falls noch kürzere Übertragungszyklen benötigt werden (z. B. das Abspeichern von Information in einen Massenspeicher), diesen als einzigen Hörer zu adressieren, so daß die Möglichkeit ausgenutzt wird, die Signale NRFD und NDAC mit Gegentakt-Ausgängen zu schalten (Bild 1, 3. Beispiel).

Bild 5. Versuchsaufbau zur Bestimmung der maximalen Datenübertragungsrate zwischen Steuergerät (links) und Communicator (rechts)

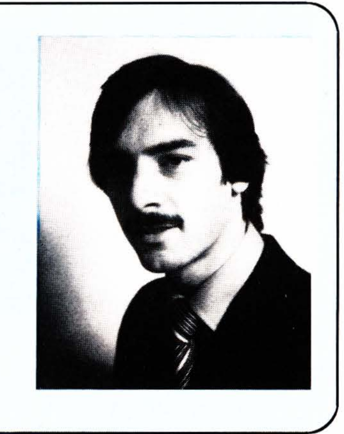
Literatur

[1] *Klein, P. E.* und *Wilhelmy, H. J.*: Der IEC-Bus (REPORT). ELEKTRONIK 1977, H. 10, S. 63...74.

[2] *Klaus, J.*: Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.

[3] Entwurf DIN IEC 66, 22. Januar 1976. DIN Deutsches Institut für Normung e. V.

Ing. (grad.) Dietmar Ehnert ist gebürtiger Kölner. Er studierte an der Gesamthochschule Paderborn und lebt heute in Köln. Die vorliegende Untersuchung führte er – in wesentlich ausführlicherer Form – als Praktikum bei der Bayer AG in Leverkusen durch; sie diente ihm gleichzeitig als Thema seiner Ingenieurarbeit. Hobbys: Fotografieren, Wandern. Privattelefon: (02 21) 78 46 16
ELEKTRONIK-Leser seit 1972.



Dr. Walter Haupt
Ing. (grad.) Wolfgang Otternberg

Zur Verlängerung des IEC-Bus: Bidirektionaler Koppler für Datenleitungen

Der Anwendungsbereich des IEC-Bus kann bei Erhaltung seiner vollen Funktionen durch Verwendung anderer Pegel erheblich erweitert werden. Zur Umsetzung auf diese Pegel wird ein bidirektionaler Koppler benutzt, dessen Funktionen im folgenden beschrieben werden. Die Verfasser haben derartige Koppler zur Verlängerung des IEC-Bus realisiert und im Betriebseinsatz erprobt. Weitere Vorteile des Kopplers sind die galvanische Trennung und eine Erhöhung der Ausgangslast.

1 Anforderungen und Realisierungsmöglichkeiten

Ein bidirektionaler Koppler muß ein aus beliebiger Richtung ankommendes Signal durchlassen. Intern wird er für jede Datenleitung je einen unidirektionalen Kanal pro Signalrichtung haben *(Bild 1)*. Um einen Selbsthalteeffekt zu vermeiden, darf nur der interne Kanal in der jeweils gewünschten Signalrichtung freigegeben werden; der entgegengerichtete Kanal muß währenddessen gesperrt sein. Die Problematik eines bidirektionalen Kopplers liegt in der Richtungssteuerung der internen Datenkanäle.

Für busfähige Systeme kommen zwei Arten der Signaldarstellung in Frage:

1. Logisch 0 und 1 werden aktiv gesendet; der passiv gesendete Zustand ist indifferent (Tristate-Technik).
2. Logisch 1 wird aktiv gesendet, logisch 0 passiv (Open-Kollektor-Technik).

Erfolgt die Richtungssteuerung im Koppler aus dem jeweiligen Signal selbst, so muß bei erstgenannter Signaldarstellung eine sichere Erkennung der 0- und 1-Signale und deren Unterscheidung vom indifferenten Zustand möglich sein. Bei der zweitgenannten Signaldarstellung genügt die eindeutige Erkennung des aktiv gesendeten 1-Signals. Selbstverständlich ist auch eine Richtungssteuerung aus einem geeigneten Begleitsignal möglich.

Steuert man die Signalrichtung im Koppler durch den Pegel des ankommenden Signals und sperrt gleichzeitig die Gegenrichtung, so wird statisch eine Selbsthaltung vermieden. Je nach Art der Schaltung neigt ein solcher Koppler wegen der internen Signallaufzeiten jedoch beim Ein- oder Abschalten des Signals zum Schwingen. Man kann dies durch geeignete Verzögerungen unterbinden, setzt jedoch dadurch auch die maximal übertragbare Signalrate herab. Steuert man die Signalrichtung ein- oder beidseitig aus der vorderen Flanke des ankommenden Signals,

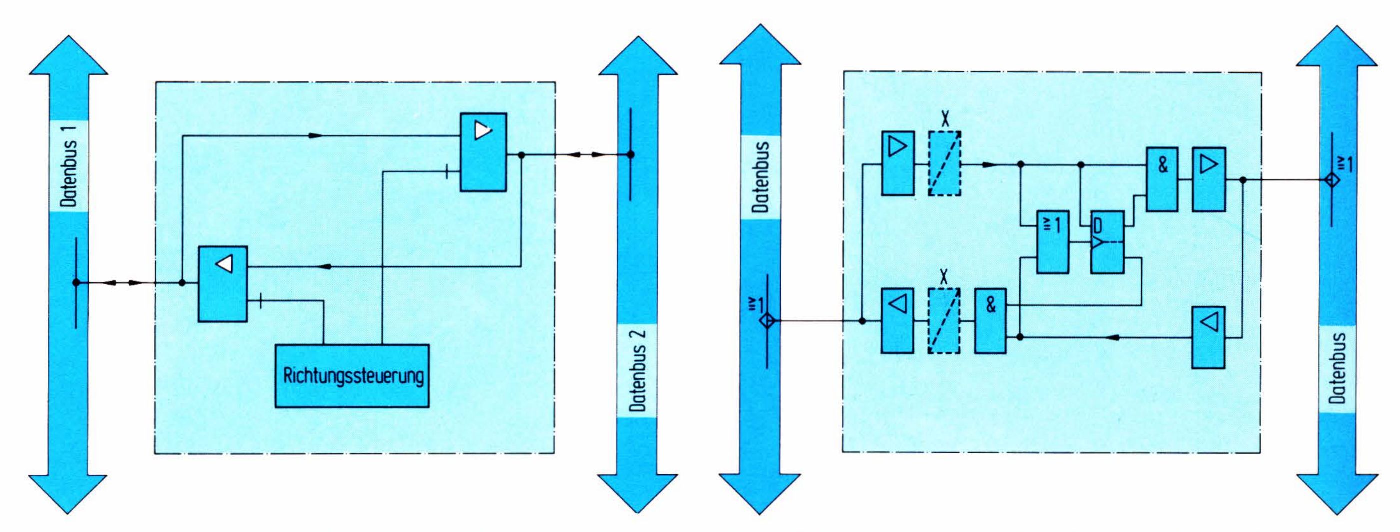


Bild 1. Blockschaltung eines bidirektionalen Kopplers für einen Datenkanal; zu den verwendeten Schaltzeichen siehe [5] und [6]

Bild 2. Bidirektionaler Koppler für einen Datenkanal mit flankengetriggerter Steuerung; zu den Schaltzeichen siehe [5] und [6]

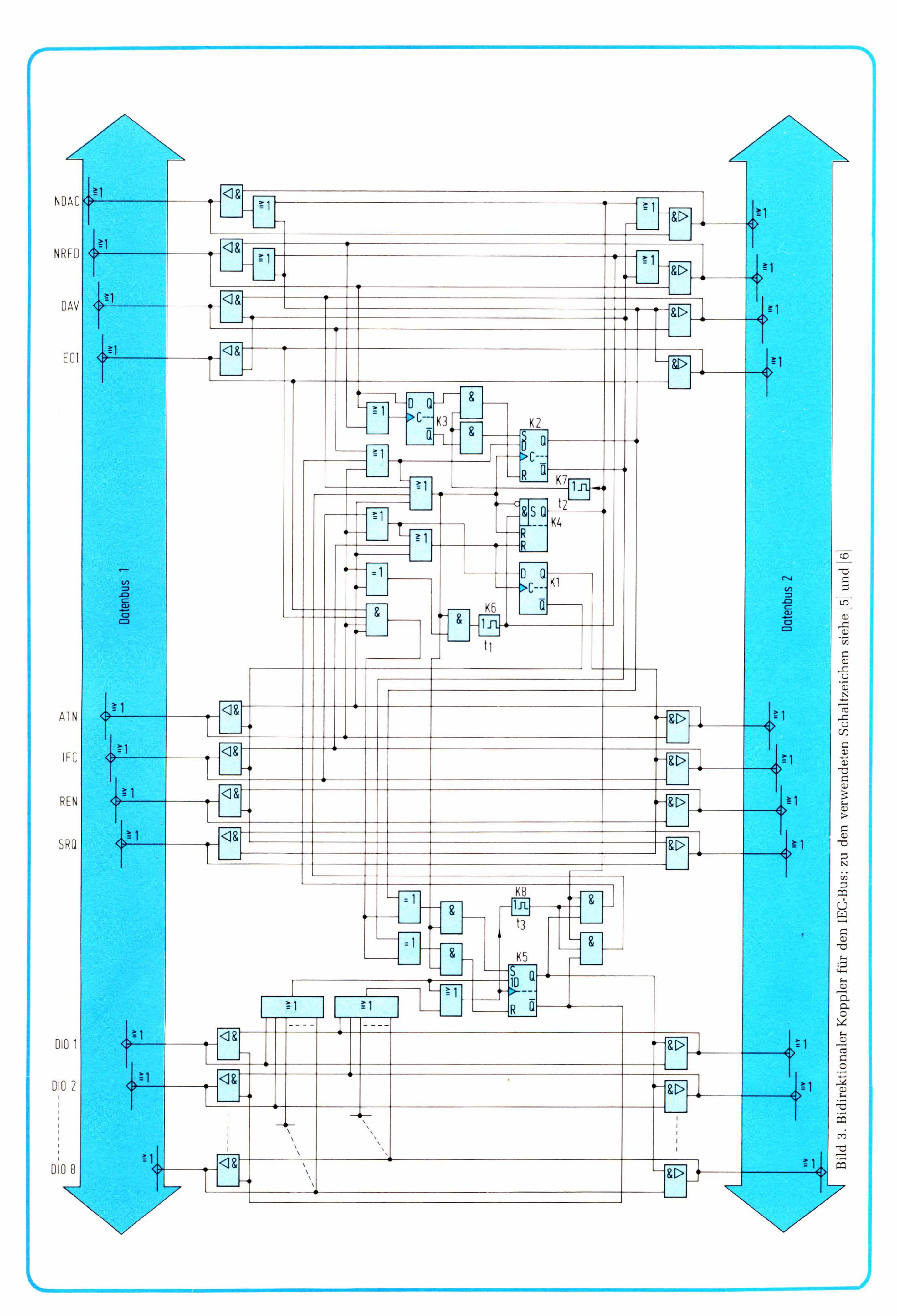


Bild 3. Bidirektionaler Koppler für den IEC-Bus; zu den verwendeten Schaltzeichen siehe [5] und [6]

so kann man die unerwünschte Selbsterregung vermeiden.

Bild 2 zeigt einen Koppler mit einer derartigen flankengetriggerten Steuerung für einen Datenkanal unter Anwendung der zweitgenannten Signaldarstellung. Die Richtungssteuerung erfolgt nach beiderseits passiv gesendetem Nullsignal beim Eintreffen des 1-Signals auf dem freizugebenden internen Kanal. Die Richtungserkennung basiert auf der internen Signaldurchlaufzeit durch den zugeordneten Bustreiber. Die Steuerung des D-Kippglieds erfolgt unter der Annahme, daß infolge der Signaldurchlaufzeit durch das dem Takteingang vorgeschaltete Oder-Gatter das Setzen des D-Eingangs beim Takten bereits erfolgt ist. Falls eine galvanische Trennung der beiden Datenbusse gewünscht wird, kann dies an den mit x bezeichneten Stellen mittels Optokopplern erfolgen.

Ein Koppler für einen bitparallelen – byteseriellen Datenbus ist realisierbar, indem man für jede Datenleitung eine Schaltung nach Bild 2 nimmt. Abgesehen von dem großen Aufwand einer jeweils separaten Richtungssteuerung pro Datenleitung scheint dies eine vernünftige Lösung zu sein. Sie setzt jedoch voraus, daß vor jeder notwendigen Richtungsumsteuerung beiderseits passiv ein Nullsignal gesendet wird. Dies kann jedoch nicht gewährleistet werden, wenn auf Datenleitungen eine Phantom-ODER-Verknüpfung erfolgt. Dies ist jedoch beim IEC-Bus für die Leitungen NRFD, NDAC und SRQ in der Norm festgelegt. Deshalb sollte man bei einem bidirektionalen Koppler für den IEC-Bus eine Richtungssteuerung nicht jeweils signalbezogen für jede Signalleitung verwenden, sondern die Signalrichtungen gruppenweise steuern für Gruppen von Signalen, die gleichen Kriterien genügen.

2 Bidirektionaler Koppler für den IEC-Bus

Beim IEC-Bus gilt für die Signale ATN, IFC, REN und SRQ das gleiche Richtungskriterium: die Lage des Steuergeräts bezogen auf den Koppler. Wird aus einem der Signale ATN oder IFC die Richtung ermittelt, so gilt sie für diese beiden Signale sowie für REN; die Signalrichtung für SRQ ist entgegengesetzt. Eine Richtungsänderung kann erst bei einer Übergabe der Steuerfunktion erfolgen. Sie wird dann jedoch gleich wieder aus einem der Signale ATN oder IFC erkannt.

Beim Datentransfer mit Handshake sind die Signalrichtungen für DAV, DIO 1...8, EOI und ATN gleich, NRFD und NDAC diesen entgegengesetzt. Adressiert das Steuergerät ein anderes Gerät, so bestimmt ATN die Signalrichtung. Übergibt das Steuergerät die Sprecherfunktion an ein anderes Gerät, so kann die Signalrichtung aus einer der Datenleitungen DIO 1...8 bestimmt werden. Dies reicht bereits für den normalen Anwendungsfall. Wird jedoch eine codetransparente Datenübertragung gewünscht, z. B. bei Benutzung des IEC-Bus zum Anschluß einer Prozeßperipherie an einen Prozeßrechner, so kann es möglich sein, daß das erste Datum ein Nullbyte ist. In diesem Fall ist aus den Datenleitungen kein Richtungskriterium zu entnehmen; es sind jedoch zwei weitere Kriterien anwendbar: aktiv gesendetes NRFD zeigt die Lage des angesprochenen Hörers an; sollte dieser jedoch bereits passiv RFD senden, so bestimmt DAV seitens des Sprechers die Richtung. Auf diese Weise kann kein Datenverlust erfolgen. Eine Besonderheit stellt die Parallelabfrage dar. Abgeleitet aus den Signalen ATN und EOI wird die Richtung der Kanäle DIO 1...8 der des ATN-Kanals entgegengeschaltet.

Bild 3 zeigt den Logikplan eines IEC-Bus-Kopplers zur Verbindung zweier Datenbusse nach der Spezfikation DIN IEC 625 [1]. Auf den Busleitungen wird logisch 1 aktiv, logisch 0 passiv gesendet. Die Richtungssteuerung der internen Kanäle im Koppler erfolgt gruppenweise durch die bistabilen Kippglieder K1, K2 und K5, deren Ausgänge die jeweils in der gewünschten Signalrichtung arbeitenden Bustreiber freigeben und die entgegenliegenden sperren.

Das Kippglied K1 bestimmt die Richtung für die Steuerleitungen ATN, IFC, REN und SRQ, das Kippglied K2 für die übrigen Steuerleitungen DAV, NRFD, NDAC und EOI. Die Steuerung für die Datenleitungen erfolgt durch das Kippglied K5. Gesetzt wird das Kippglied K1 nach beidseitig passiv gesendetem Nullsignal vom ersten 1-Signal auf einer der Leitungen ATN oder IFC. Dieser Zustand bleibt erhalten, solange keine Übergabe der Steuerfunktion an ein Gerät erfolgt, das an dem Datenbus auf der anderen Seite des Kopplers angeschlossen ist. Der Zustand des Kippglieds K3 wird durch das Signal NRFD bestimmt. Dieses Kippglied steuert keine Signalrichtung unmittelbar.

Das Kippglied K5 wird nach beidseitigem Nullbyte auf dem Datenleitungen DIO 1...8 beim Eintreffen einer Nachricht auf diesen Leitungen in Richtung dieser Nachricht gesetzt, sofern weder die Nachricht ATN noch DAV wahr gesendet wird. Wenn ATN oder DAV wahr gesendet wird, bestimmt der Zustand des Kippglieds K2 den Zustand von K5.

Der Zustand von K2 wird primär nach beidseitig passiv Null gesendeten Nachrichten DAV und ATN aus der zuerst wahr gesendeten Nachricht DAV oder ATN bestimmt. Eine Korrektur dieses Zustands aufgrund anderer Kriterien kann erfolgen nach dem Übergang der Nachricht ATN = 1 auf ATN = 0. Wenn dieser Übergang erfolgt, kann die Sprecherfunktion vom Steuergerät an ein vorher entsprechend adressiertes Gerät übergeben werden. In diesem Fall kann eine Richtungsumkehr der Signale erfolgen, die der Koppler aus der Nachricht DAV zu spät erkennt. Aus diesem Grund werden beim Übergang von ATN = 1 auf ATN = 0 für die Zeit t_1, ausgelöst durch das monostabile Kippglied K6, für das Signal NRFD beide interne Kanäle freigegeben. Liegt das Signal NRFD = 1 an, so geht für die Zeit t_1 der Koppler für dieses Signal in Selbsthaltung. Dies bedingt eine Verzögerung des Handshake-Vorgangs um maximal t_1. Die Zeit t_1 sollte größer sein als das Doppelte der längsten der Signallaufzeiten auf den Bussen beiderseits des Kopplers. Als Richtwert ist eine oder bei großen Entfernungen einige µs anzugeben.

Beim Übergang von ATN = 1 auf ATN = 0 wird das Kippglied K4 gesetzt, welches beide Richtungen für den Kanal NDAC freigibt, so daß der Handshakevorgang nicht beendet werden kann, wenn ein Gerät NDAC wahr sendet oder während dieses Zustands des Kippglieds K4 gesendet hat. Nach dem Setzen von K4 wird innerhalb einer kurzen Zeit t_2, gegeben durch das monostabile Kippglied K7, der Zustand von K3 auf K2 übertragen. Wenn nach Ablauf der Zeit t_2 das Kippglied K4 noch gesetzt ist, wird beim Eintreffen eines Bytes nach passiv gesendetem Nullbyte auf den Datenleitungen DIO 1...8 der Zustand von K5 auf K2 übertragen. Anschließend wird das Kippglied K4 zurückgesetzt. Dieses Rücksetzen erfolgt ferner, wenn ein Signal DAV, ATN oder IFC wahr gesendet wird.

Bei normaler Adreß-, Befehls- oder Datenübertragung erfolgt die Ansteuerung des Kippglieds K5 durch K2 gleichsinnig. Bei der Parallelabfrage wird, initiiert durch ATN = 1 und EOI = 1, diese Ansteuerung invertiert.

Die *Tabellen 1...5* zeigen die Wahrheitstabellen für die bistabilen Kippglieder K1...K5, *Tabelle 6* die für das monostabile Kippglied K6. Die Wahrheitstabellen beziehen sich auf die Bussignale und z. T. auf die Ausgangssignale der monostabilen Kippglieder als Eingangssignale. Die horizontalen Pfeile unter den angegebenen Eingangssignalen geben die interne Signalrichtung nach Bild 3 an. Die Angabe DIO 1...8 in Tabelle 5 gilt für eine ODER-Verknüpfung dieser Signale.

IEC-Bus-Koppler, funktionell entsprechend Bild 3, wurden in der Bayer AG realisiert zur Verlängerung des IEC-Bus. Rechnerseitig arbeitete der Koppler nach den Originalspezifikationen DIN IEC 625; der anderseits angeschlossene Bus war schaltungstechnisch

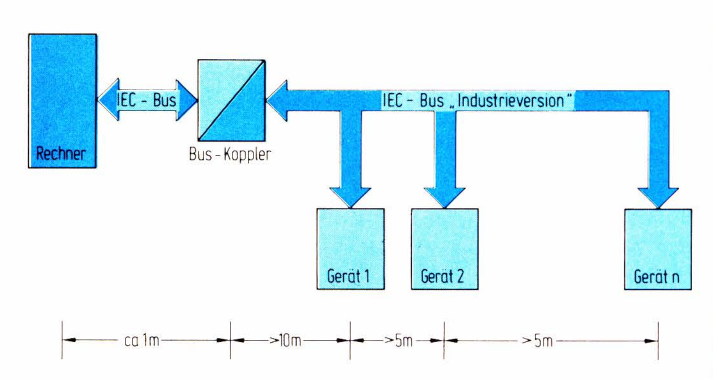


Bild 4. Anwendung des bidirektionalen Kopplers zur Erweiterung des IEC-Bus

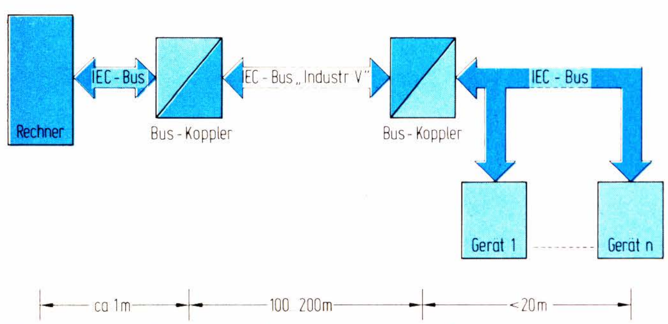


Bild 5. Anwendung des bidirektionalen Kopplers zur Verlängerung des IEC-Bus

Tabelle 1. Wahrheitstabelle für das Kippglied K 1

ATN		ICF		Q_1	Q_1
→	←	→	←		
↑	0	0	0	1	0
1	x	x	x	Q_1	$\overline{Q_1}$
0	↑	0	0	0	1
x	1	x	x	Q_1	$\overline{Q_1}$
0	0	↑	0	1	0
x	x	1	x	Q_1	$\overline{Q_1}$
0	0	0	↑	0	1
x	x	x	1	Q_1	Q_1

Tabelle 2. Wahrheitstabelle für das Kippglied K 2

DAV		ATN		Q_7	Q_8	Q_2	Q_2
→	←	→	←				
↑	0	0	0	0	0	1	0
1	x	x	x	0	x	Q_2	$\overline{Q_2}$
0	↑	0	0	0	0	0	1
x	1	x	x	0	x	Q_2	$\overline{Q_2}$
0	0	↑	0	0	0	1	0
x	x	1	x	0	x	Q_2	$\overline{Q_2}$
0	0	0	↑	0	0	0	1
x	x	x	1	0	x	Q_2	$\overline{Q_2}$
x	x	x	x	1	x	$\overline{Q_3}$	Q_3
0	0	0	0	0	↑	Q_5	$\overline{Q_5}$
x	x	x	x	0	1	Q_2	$\overline{Q_2}$

Tabelle 3. Wahrheitstabelle für das Kippglied K 3

NRFD		Q_3	$\overline{Q_3}$
→	←		
↑	0	1	0
1	x	Q_3	$\overline{Q_3}$
0	↑	0	1
x	1	Q_3	$\overline{Q_3}$

Tabelle 4. Wahrheitstabelle für das Kippglied K 4

ATN	DAV	IFC	Q_6	Q_8	Q_4	$\overline{Q_4}$
0	0	0	1	0	1	0
1	x	x	x	x	0	1
x	1	x	x	x	0	1
x	x	1	x	x	0	1
x	x	x	x	1	0	1

Tabelle 5. Wahrheitstabelle für das Kippglied K 5

DIO	1–8	ATN	EOI	DAV	Q_5	$\overline{Q_5}$
→	←					
↑	0	0	x	0	1	0
1	x	0	x	0	Q_5	$\overline{Q_5}$
0	↑	0	x	0	0	1
x	1	0	x	0	Q_5	$\overline{Q_5}$
x	x	1	0	x	Q_2	$\overline{Q_2}$
x	x	1	1	x	$\overline{Q_2}$	Q_2
x	x	x	0	1	Q_2	$\overline{Q_2}$

Tabelle 6. Wahrheitstabelle für das Kippglied K 6

ATN		Q_6
→	←	
1	↓	Ω
↓	1	Ω

nach der Spezifikation CCITT V 11 ausgelegt [1, 2]. Es wurden Differenztreiber in Tristatetechnik verwendet, bei denen die Signaleingabe über den Steuereingang erfolgt [3, 4]. Dadurch wird logisch 1 aktiv, logisch 0 passiv gesendet. Die beiden Datenbusse wurden durch Optokoppler galvanisch entkoppelt, die entsprechend Bild 2 angeordnet waren.

Die *Bilder* 4 und 5 zeigen realisierte Anwendungen des Kopplers zur Verlängerung des IEC-Bus.

3 Ausblick

Ein bidirektionaler Koppler für den IEC-Bus vermag den durch die Norm DIN IEC 625 gegebenen Anwendungsbereich hinsichtlich der Ausdehnung des Bussystems und der Anzahl der anzuschließenden Geräte zu erweitern. Einer einheitlichen und breiten Anwendung steht die fehlende entsprechende Norm entgegen. Es ist jetzt an der Zeit, für den in CCITT V 11 erwähnten Mehrpunktbetrieb den Kabelabschluß und eine Vorspannungserzeugung für einen Busbetrieb mit einer Phantom-ODER-Verknüpfung auf Datenleitungen zu normen.



Ing. (grad.) Wolfgang Otternberg ist alteingesessener Leverkusener. Er studierte Physikalische Technik in Iserlohn. Seit 1967 ist er bei der Bayer AG tätig in der elektronischen Geräteentwicklung (Bussysteme).
Hobbys: Musizieren, Fotografieren, soweit zwei Kinder Zeit dazu lassen.
ELEKTRONIK-Leser seit 1967

Dr. rer. nat. Walter Haupt ist gebürtiger Bonner mit Studium der Physik in Bonn. Seit 1960 ist er bei der Bayer AG tätig und u. a. Mitglied in den DKE-Ausschüssen K 923 „Elektronische Meßgeräte"; UK 933.1 „Meßgeräte – Schnittstellen".
Hobbys: Klassische Musik, Wandern.
ELEKTRONIK-Leser seit 1957

Literatur

[1] DIN IEC 625 Elektrische Meßtechnik. Byteserielles, bitparalleles Schnittstellensystem für programmierbare Meßgeräte. Z. Z. in Druckvorbereitung. Siehe Entwurf DIN IEC 66.22 Beuth-Verlag GmbH, Berlin, Köln.

[2] CCITT-Empfehlungen der V-Serie und der X-Serie: Datenübertragung. 3. Auflage 1977. R. v. Decker's Verlag, G. Schenk, Heidelberg, Hamburg.

[3] *Ehnert, D.*: Das Zeitverhalten einer IEC-Bus-Version für größere Entfernungen. ELEKTRONIK 1978, H. 4, S. 62...64.

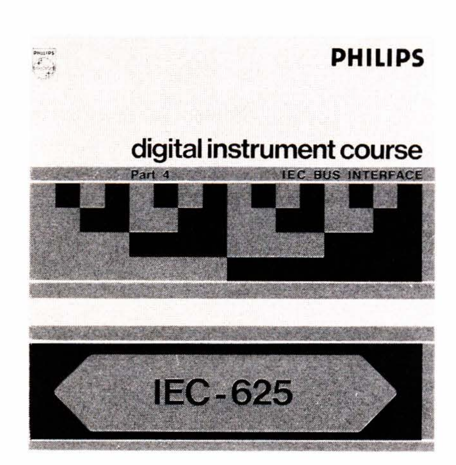
[4] *Haupt, W.*; *Otternberg, W.*: Der IEC-Bus für größere Entfernungen. ELEKTRONIK 1979, H. 7, S. 74...76.

Zum Nachschlagen betreffend die vom Autor verwendeten neuen Schaltzeichen empfiehlt die Redaktion:

[5] *Bürgel, E.*: Neue Normen und Schaltzeichen der digitalen Informationsverarbeitung. Franzis-Verlag, München, 1978.

[6] Neue Schaltzeichen der Digitaltechnik. ELEKTRONIK-Arbeitsblatt Nr. 109, Teil 1...3. ELEKTRONIK 1978, H. 1, S. 85...86, H. 2, S. 75...78, H. 3, S. 85...88.

IEC-Bus: Ein Intensiv-Training



Im Vertrieb der Philips GmbH, Unternehmensbereich Elektronik für Wissenschaft und Industrie, befindet sich ein äußerst interessantes Buch, das sich vor allem für die fundierte Intensiv-Einarbeitung in die Thematik des IEC-Bus eignet.

Verfaßt in englischer Sprache bringt es auf 112 Seiten und unterstützt durch zahlreiche Tabellen, Schaltbilder und Zustandsdiagramme einen umfassenden Überblick nicht nur über den grundsätzlichen Aufbau des Bus-Systems, vielmehr werden auch die Signalzustände und -abhängigkeiten, die Adressierung und die System-Implementierung eines Bus-Interface beleuchtet. Den Abschluß bilden praktisch ausgeführte Applikationen sowie ein Fragen- und Antwort-Anhang, der zur Überprüfung des kapitelweise erarbeiteten Wissens dient.

Hier der genaue Titel: Philips Digital Instrument Course, Part 4 – IEC-Bus-Interface.

Vertrieb durch die Philips GmbH, Unternehmensbereich für Wissenschaft und Industrie, Postfach 31 03 20, 3500 Kassel, Preis etwa 15.– DM. *Ha*

So einfach für den Anwender die Verkettung fertiger Meßgeräte, die mit IEC-Bus-Anschluß ausgerüstet sind, zu kompletten Systemen ist, so schwierig ist für den Geräteentwickler das Anpassen seiner Schaltungen an den IEC-Bus über eine normgerechte Interface-Schaltung. Gerade dieser Entwicklungsschritt wird hier in genereller Form und anhand eines konkreten Beispiels beschrieben. Dabei weicht die Bezeichnungsweise für den Bus beim Autor aus firmeninternen Gründen etwas von der in Deutschland eingeführten und von der ELEKTRONIK bisher benutzten ab; er spricht nicht vom IEC-Bus, sondern von Interface-Bus. Dies dürfte jedoch keine Verwirrung stiften, und so sprechen wir in der Überschrift, wie gewohnt, vom IEC-Bus, während im Text die vom Autor benutzte Bezeichnung „Interface-Bus" beibehalten wurde.

Dipl.-Ing. Arndt Pannach

Interface-Entwicklung für den IEC-Bus

1 Die Interface-Funktionen

Die funktionellen Elemente eines an den Interface-Bus angeschlossenen Gerätes sind in *Bild 1* aufgezeigt.

Wie der IEC-Bus funktioniert, wurde in der ELEKTRONIK eingehend erklärt [6]. Dennoch dürfte die folgende kurze Zusammenfassung von Nutzen sein:

Der Interface-Bus setzt sich zusammen aus dem Daten-Bus (8 Leitungen), dem Handshake-Bus (3 Signale) und dem Steuer-Bus (5 Signale). Adressen, Befehle, Programmier- und Meßdaten werden asynchron im Dreidraht-Handshake-Verfahren übertragen. Die Übertragung verläuft bit-parallel und zeichen-seriell.

Senden, Empfangen und Verarbeiten von Bus-Nachrichten erfolgt durch die Interface-Funktionen; sie bilden daher den wichtigsten Teil einer Interface-Schaltung. Jede Interface-Funktion hat eine ihr zugewiesene spezifische Aufgabe (*Tabelle 1*), deren Erledigung nach einem definierten Protokoll zu geschehen hat. Das zu jeder Interface-Funktion gehörende Protokoll ist in der Norm in Form von *State-Diagrammen* [8] dargestellt; diese bilden die wichtigste Grundlage für den Schaltungsentwurf.

Tabelle 1. Abkürzungen für die Interface-Funktionen

Abkürzung	Interface-Funktion	
SH	Handshakequelle	(Source Handshake)
AH	Handshakesenke	(Acceptor Handshake)
T	Sprecher	(Talker)
L	Hörer	(Listener)
SR	Bedienungsruf	(Service Request)
RL	Fern/eigen-Umschaltung	(Remote Local)
PP	Parallelabfrage	(Parallel Poll)
DC	Gerät rücksetzen	(Device Clear)
DT	Gerät auslösen	(Device Trigger)
C	Steuereinheit	(Controller)

Neben den funktionellen Elementen sind in Bild 1 auch die verschiedenen *Signalwege* und *Nachrichtenarten* dargestellt. Interne Nachrichten werden zwischen den Interface-Funktionen und den Gerätefunktionen ausgetauscht.
Externe Nachrichten werden über den Interface-Bus übertragen. Man unterscheidet zwei Arten:
(1) Nachrichten, die von einer Interface-Funktion zu einer anderen gesendet werden. Beispiel: Die C-Funktion eines

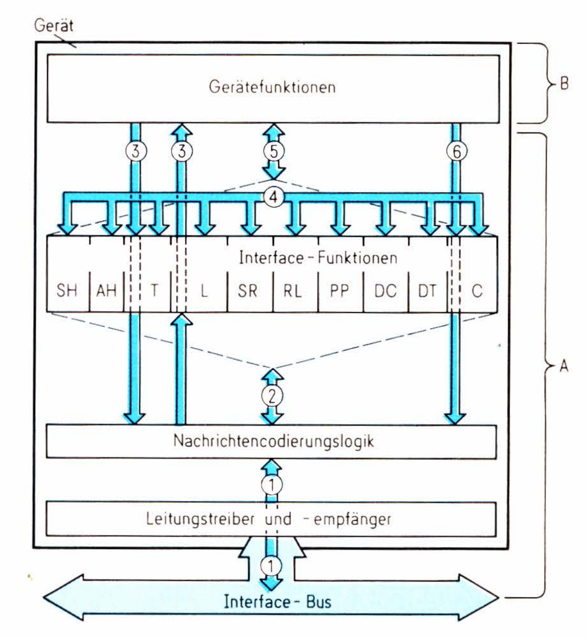
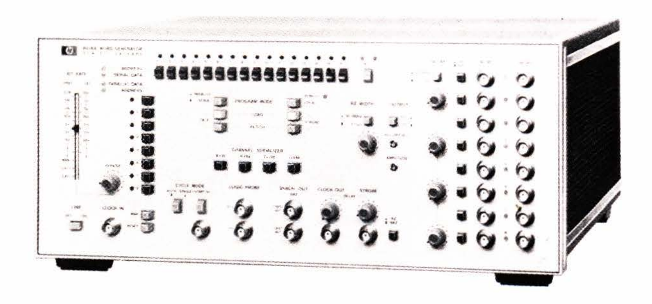
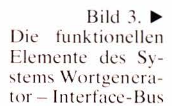


Bild 1. Die funktionellen Elemente des Systems Gerät-Interface-Bus. Anmerkungen:
A Fähigkeiten sind in der vorliegenden Norm festgelegt.
B Fähigkeiten werden vom Entwickler festgelegt. Es bedeuten:
(1) Signalleitungen im Interface-Bus
(2) Externe Nachrichten an und von Interface-Funktionen
(3) Gerätenachrichten an und von Gerätefunktionen
(4) Zustandsverknüpfungen zwischen Interface-Funktionen
(5) Interne Nachrichten zwischen Geräte-Funktionen und Interface-Funktionen
(6) Externe Nachrichten, die nur innerhalb eines Steuergerätes von Gerätefunktionen gesendet werden





◀ Bild 2. Wortgenerator 8016 A von Hewlett-Packard

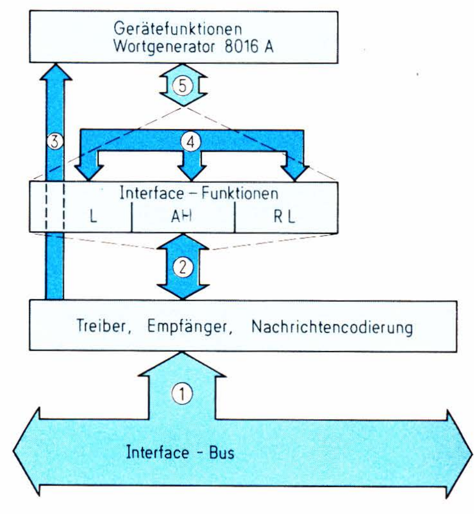


Bild 3. ▶ Die funktionellen Elemente des Systems Wortgenerator – Interface-Bus

Tischrechners sendet auf den Interface-Bus eine Listener-Adresse. Empfänger ist die Listener-Funktion des Gerätes, das adressiert werden soll.

(2) Nachrichten, die zwischen den Gerätefunktionen ausgetauscht werden. Beispiel: Ein Digitalvoltmeter sendet Meßdaten zu einem Drucker.

2 Vier Schritte zur Interface-Schaltung

Die Entwicklung einer Interface-Schaltung nach dem IEC-Normenentwurf vollzieht sich in vier Schritten. Der *erste Schritt* beinhaltet die Erarbeitung der eigentlichen Zielsetzung. Diese umfaßt eine genaue Beschreibung der Aufgaben, die das Gerät in einem Bus-System wahrzunehmen hat. Die Art des Gerätes sowie seine künftige Anwendung werden hierfür bestimmend sein.

In einem *zweiten Schritt* werden die zu realisierenden Interface-Funktionen festgelegt. Der Normenentwurf offeriert zehn Interface-Funktionen, aus denen gemäß der Zielsetzung die richtige Auswahl zu treffen ist. Einen Drucker oder eine Lochstreifenstanze wird man beispielweise mit der L-, der AH- und der RL-Interface-Funktion ausstatten. Ein Digitalvoltmeter wird zusätzlich die T- und die SH-Interface-Funktion benötigen. Die C-Interface-Funktion wird normalerweise in einem Tischrechner oder Computer zu finden sein.

Der *dritte Schritt* besteht aus dem eigentlichen Schaltungsentwurf. Eine einheitliche Lösung gibt es naturgemäß nicht. Wahl der Logikfamilie, wirtschaftliche Erwägungen und Kenntnisstand des Entwicklers sind die bestimmenden Faktoren. Später soll am Beispiel des Wortgenerators HP 8016A ein möglicher Schaltungsentwurf kurz aufgezeigt werden.

Schließlich muß im *vierten Schritt* die Funktionsweise überprüft werden. Um sicherzustellen, daß das Gerät in jeder beliebigen Systemkonfiguration arbeitet, sollte die erste Erprobung unabhängig von einem bestimmten System erfolgen. Für diese Aufgabe werden bereits spezielle Prüfgeräte angeboten. Es folgt die endgültige Erprobung in einem Interface-Bus-System.

3 Beispiel: Interface-Funktionen eines Wortgenerators

Bei dem Gerät, das hier als Beispiel für den Entwurf einer Bus-Schnittstelle dient, handelt es sich um einen Bitmuster- oder Wortgenerator (*Bild 2*). Das Gerät hat acht Ausgangskanäle. Jeder Kanal liefert 32 Bit mit einer maximalen Frequenz von 50 MHz. Der Umschalter „Channel Serializer" erlaubt serielle Datenfolgen bis maximal 256 Bit.

Wortgeneratoren haben folgende Einsatzmöglichkeiten:

– Testen von komplexen digitalen ICs;
– Simulieren von Datenquellen und Austesten von Datenempfängern;
– Austesten von Datenbussen;
– Lieferung des Testsignals für Baugruppen der PCM- und Datenübertragungstechnik.

Der zentrale Schaltungsteil des vorliegenden Wortgenerators ist ein Schreib-Lese-Speicher (RAM); in ihn wird das binäre Prüfmuster manuell geladen. Für Aufgaben, bei denen das Prüfmuster häufig und schnell geändert werden muß, wie z. B. im Prüffeld oder in System-Anwendungen, können die Daten über den Interface-Bus geladen werden. Als Steuergeräte dienen Tischrechner oder optische Kartenleser. Um diese Aufgabe wahrnehmen zu können, mußte für den Wortgenerator die L-, die AH- und die RL-Interface-Funktion ausgewählt werden (*Bild 3*).

Der Code für Adressen und für einen Teil der Befehle ist in der Norm vorgeschrieben. Die eigentliche Datenübertragung hingegen ist völlig codetransparent. Für den Wortgenerator ist es sinnvoll, die Daten rein binär zu übertragen. Geräte, wie Digitalvoltmeter oder Pulsgeneratoren, werden in der Regel den ISO-7-Bit-Code (ASC II-Code) verwenden.

4 Das L- und AH-State-Diagramm

Das State Diagramm der L-Funktion ist in *Bild 4* dargestellt; sie kann drei verschiedene Zustände (States) einnehmen, die durch drei Kreise gekennzeichnet sind. Jeder Zustand hat einen Namen, dessen mnemotechnische Abkürzung in jedem Kreis angegeben ist. Alle erlaubten Übergänge von einem Zustand zu einem anderen sind durch Pfeile gekennzeichnet; jedem Pfeil ist ein Boolescher Ausdruck beigefügt, der wahr oder falsch sein kann. Ist der Ausdruck wahr, muß der Übergang zu einem anderen Zustand in Pfeilrichtung

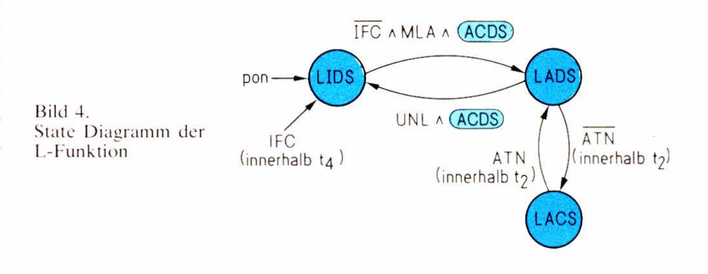
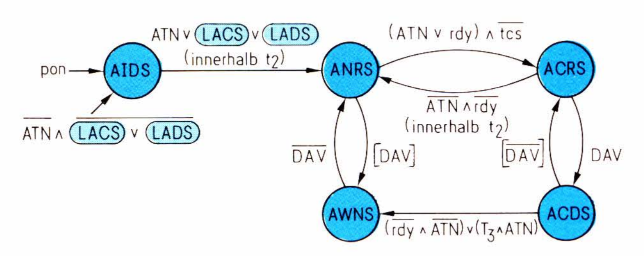


Bild 4. State Diagramm der L-Funktion



◀ Bild 5. State Diagramm der AH-Funktion

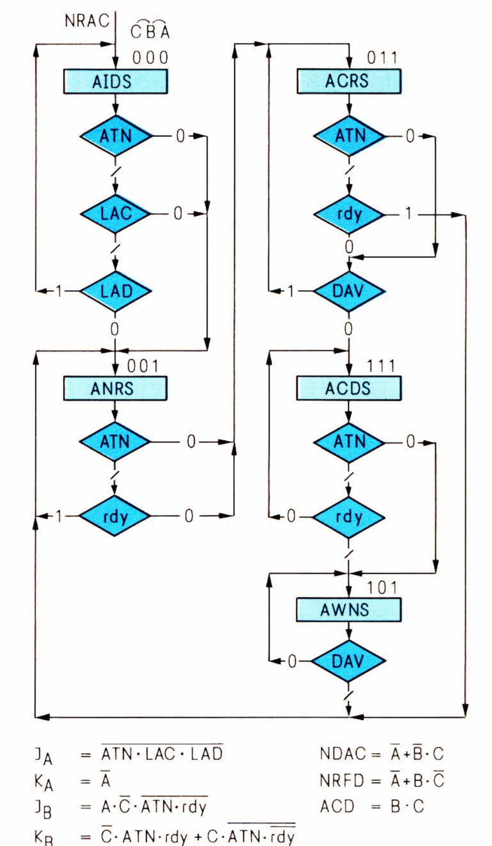


Bild 6. ▶ Modifiziertes State-Diagramm der AH-Funktion mit Eingangs- und Ausgangsgleichungen

$J_A = \overline{ATN \cdot LAC \cdot LAD}$

$K_A = \overline{A}$

$J_B = A \cdot \overline{C} \cdot \overline{ATN \cdot rdy}$

$K_B = \overline{C} \cdot ATN \cdot rdy + C \cdot \overline{ATN \cdot \overline{rdy}}$

$J_C = B \cdot \overline{DAV} \cdot \overline{ATN \cdot rdy}$

$K_C = \overline{B} \cdot DAV$

$NRAC = \overline{\overline{pon} + ATN \cdot LAC \cdot LAD}$

$NDAC = \overline{A} + \overline{B} \cdot C$

$NRFD = \overline{A} + B \cdot \overline{C}$

$ACD = B \cdot C$

stattfinden; ist er falsch, muß der derzeitige Zustand beibehalten werden. Es gibt auch Pfeile, die ohne Ursprung auf einen bestimmten Zustand zeigen (LIDS in Bild 4); sie zeigen an, daß der Übergang von jedem beliebigen Zustand der Interface-Funktion auf den durch den Pfeil gekennzeichneten Zustand erfolgen muß, wenn der zugehörige Boolesche Ausdruck wahr ist.

Die Booleschen Ausdrücke bestehen aus:

- *Nachrichten von Geräte-Funktionen.* Diese sind gekennzeichnet durch drei kleine Buchstaben, z. B. pon (power on – Gerät eingeschaltet) in Bild 4.
- *Nachrichten von anderen Interface-Funktionen.* Die mnemotechnische Abkürzung besteht aus drei großen Buchstaben. Beispiel in Bild 4: MLA (My Listen Address – eigene Höreradresse).
- *Zustandsverknüpfungen.* Diese sind gekennzeichnet durch vier große Buchstaben, umschlossen von einer Ellipse. Beispiel in Bild 4: ACDS (Accept Data State-Datenübernahmezustand). Diese Zustandsverknüpfungen entsprechen dem Signalweg 4 in Bild 1.
- *Zeitoperatoren.* Ein kleines t besagt, daß der Übergang zum nächsten Zustand innerhalb dieser Zeit erfolgen muß. Ein großes T zeigt an, daß der jetztige Zustand für mindestens diese Zeit beibehalten werden muß. Beispiel in Bild 4 : t_2

Die L-Funktion (Bild 4) hat drei Zustände:

- den Zustand „*Hörer in Ruhe*" (LIDS-Listener Idle State)
- den Zustand „*Hörer adressiert*" (LADS-Listener Addressed State)
- den Zustand „*Hörer aktiv*" (LACS-Listener Active State)

In LIDS und LADS können von der L- Funktion Adressen und Befehle empfangen werden. Diese Nachrichten werden von den Gerätenachrichten dadurch unterschieden, daß das Signal ATN (Attention-Achtung) wahr ist. Der Empfang von Gerätenachrichten, also von Meß- und Programmierdaten, ist nur dann möglich, wenn sich die L-Funktion in LACS befindet.

Die L-Funktion geht von LIDS nach LADS, wenn die Höreradresse (MLA) empfangen wird *und* die AH-Funktion die Gültigkeit der Daten meldet (ACDS) *und* das Signal IFC (Interface Clear-Interface zurücksetzen) falsch ist. Der Übergang von LADS nach LACS erfolgt, wenn ATN falsch wird.

Die AH-Funktion sorgt für einen geordneten Empfang von externen Nachrichten nach den Regeln des Dreidraht-Handshake-Verfahrens. Der asynchrone Handshake-Zyklus wird zusammen mit der SH-Funktion über den Handshake-Bus abgearbeitet; dieser führt die folgenden Signale:
- DAV (Data Valid – Daten gültig)
- NRFD (Not Ready For Data – nicht bereit für Daten)
- NDAC (Not Data Accepted – Daten nicht übernommen)

Die logischen Zustände der Signale NRFD und NDAC werden von der AH-Funktion gesteuert (*Tabelle 2*); diese Funktion (*Bild 5*) hat fünf Zustände: Nach dem Einschalten des Gerätes soll sie in den Zustand AIDS gehen. Falls ATN wahr ist *oder* die L-Funktion adressiert ist (LADS) *oder* die L-Funktion aktiv ist (LACS), muß der Übergang von AIDS nach ANRS erfolgen, und zwar innerhalb t_2. ACRS wird erreicht wenn ATN wahr ist oder rdy (ready for next message – bereit für nächste Nachricht) wahr ist. Die Nachricht rdy si-

Tabelle 2. Bezeichnungen der Zustände im AH-State-Diagramm

AH-State	Bezeichnung	NRFD	NDAC
AIDS	Zustand „Handshakesenke in Ruhe" (Acceptor Idle State)	HIGH	HIGH
ANRS	Zustand „Handshakesenke nicht bereit" (Acceptor Not Ready State)	LOW	LOW
ACRS	Zustand „Handshakesenke bereit" (Acceptor Ready State)	HIGH	LOW
ACDS	Zustand „Handshakesenke übernimmt Daten" (Accept Data State)	LOW	LOW
AWNS	Zustand „Handshakesenke wartet auf neuen Zyklus" (Acceptor wait for new Cycle State)	LOW	HIGH

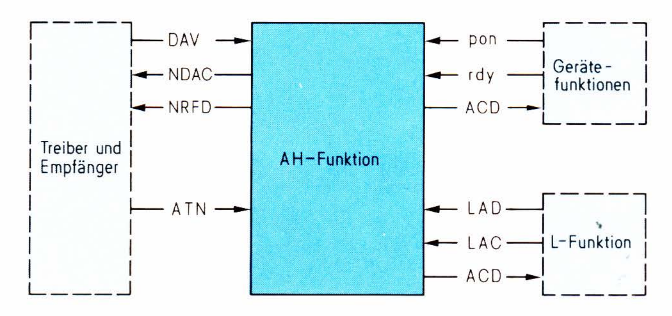


Bild 7. Eingangs- und Ausgangssignale der AH-Funktion

gnalisiert der AH-Funktion die Bereitschaft der Gerätefunktionen zum Empfang von Gerätenachrichten. Wird DAV wahr, findet der Übergang nach ACDS statt; dieser Zustand zeigt der L-Funktion und den Gerätefunktionen an, daß auf dem Daten-Bus gültige Daten sind. Die Gerätefunktionen müssen zudem die L-Funktion abfragen, denn nur wenn diese sich im Zustand LACS befindet, handelt es sich um Gerätenachrichten. Die AH-Funktion geht von ACDS nach AWNS, falls ATN wahr ist und eine Zeit T_3 verstrichen ist *oder* wenn ATN falsch ist und rdy falsch ist. T_3 ist abhängig vom jeweiligen Entwurf und muß so gewählt werden, daß die L-Funktion genügend Zeit hat, ihren Zustands-Übergang zu vollziehen. Von AWNS geht es schließlich zurück zu ANRS, falls DAV falsch wird.

Boolesche Ausdrücke, die in eckigen Klammern stehen, müssen beim Schaltungsentwurf nicht berücksichtigt werden; sie können aber verwendet werden, wenn sich dadurch zum Beispiel Vereinfachungen ergeben.

5 Die Schaltung der AH-Funktion

Von den funktionellen Elementen der Interface-Schaltung des besprochenen Wortgenerators (Bild 3) soll die AH-Funktion als Entwurfsbeispiel dienen. Die Eingangs- und Ausgangssignale – und zwar nur diejenigen, die für die AH-Funktion Bedeutung haben – sind in *Bild 7* dargestellt. Die Signale LAC und LAD sind im Zustand LACS bzw. LADS wahr; in gleicher Weise ist das Signal ACD im Zustand ACDS wahr.

Für die Realisierung der Funktion bietet sich der Entwurf eines *synchronen sequentiellen Schaltwerkes* an. Dies nicht zuletzt deshalb, weil ja das State-Diagramm ohnehin die Grundlage oder das wesentliche Hilfsmittel vieler Methoden zum Entwurf synchroner sequentieller logischer Schaltungen ist; diese werden in der Regel mit Flipflops oder mit Festwertspeichern (ROMs) realisiert.

Hier wurden JK-Flipflops gewählt, die gemeinsam getaktet werden. Zu jedem Takt-Zeitpunkt bilden die Flipflop-Ausgänge einen stabilen inneren Zustand. Mit N Flipflops sind demnach 2^n verschiedene Zustände möglich. Der Übergang von einem Zustand zu einem anderen ist abhängig von dem derzeit gültigen Zustand selbst und von den Eingangssignalen. Die Ausgänge der Flipflops steuern zusammen mit den Eingangssignalen über Logik-Schaltungen die Vorbereitungseingänge J und K, die ja letztlich für den Übergang zu einem anderen Zustand bestimmend sind. Aus dem Gesagten ergibt sich, daß die Hauptaufgabe für den Entwurf einer sequentiellen Schaltung darin besteht, die Booleschen Gleichungen für die J- und K-Eingänge der Flipflops zu finden.

Bild 6 zeigt ein modifiziertes State-Diagramm der AH-Funktion. Die Eingangssignale sind in Rauten eingezeichnet. Sie sind hier so definiert, daß sie wahr sind, wenn ihr logischer Zustand „0" ist. Die verschiedenen Zustände sind durch Rechtecke dargestellt. Das Verharren in einem Zustand oder der Übergang zu einem anderen Zustand erfolgt in Pfeilrichtung. Die sequentielle Schaltung soll fünf Zustände durchlaufen, für deren Codierung drei Flipflops mit den Ausgängen A, B, C benötigt werden. Die Codierung ist beliebig gewählt und an jeden Zustand angeschrieben.

Das Signal NRAC ist ein für alle Flipflops gemeinsames asynchrones Rücksetzsignal, das die AH-Funktion aus allen anderen Zuständen in den Ruhezustand bringt. Dieses State-Diagramm ist die Grundlage für die Ermittlung der FF-Eingangsgleichungen. Auf die nun folgenden Entwurfsschritte kann im Rahmen dieses Aufsatzes nicht näher eingegangen werden. Der interessierte Leser sei vielmehr auf die Literaturangabe [7] verwiesen. In Bild 6 sind die für J_A bis K_C gefundenen Gleichungen angegeben. Die Gleichungen für NDAC und NRFD ergeben sich direkt aus Tabelle 2 und der Codierung der Zustände. Das Ausgangssignal ACD ergibt sich aus der Codierung des Zustands ACDS. Die praktisch ausgeführte Logikschaltung für die AH-Interface-Funktion ist in *Bild 8* dargestellt.

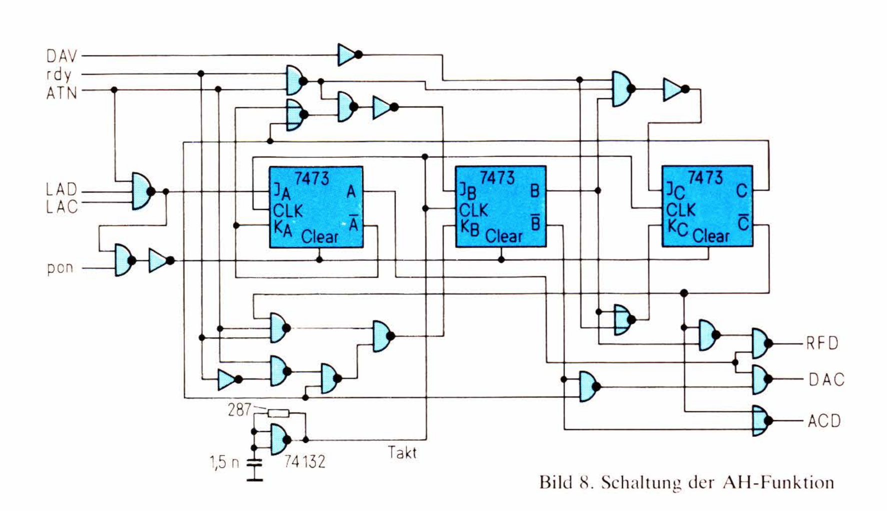


Bild 8. Schaltung der AH-Funktion

Literatur

[1] *Loughry, D.C.:* A common digital interface for programmable instruments: the evolution of a system. Hewlett-Packard Journal, Oktober 1972.

[2] *Ricci, D.W.* und *Nelson, G. E.:* Standard instrument interface simplifies system design. Electronics, Bd. (1974), H. 23, S. 95...106.

[3] *Loughry, D.C.:* The Hewlett-Packard Interface Bus: Current Perspectives. Hewlett-Packard Journal, Januar 1975.

[4] IEEE Std 488-1975: IEEE Standard Digital Interface for Programmable Instrumentation.

[5] *Knoblock, Loughry, Vissers:* Insight into interfacing. IEEE spectrum, Mai 1975, S. 50...57.

[6] *Klaus, J.:* Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.

[7] *Clare, C.R.:* Designing Logic Systems using states machines. MC Graw-Hill Inc., 1973.

[8] *Künzel, R.:* Das State-Diagramm. ELEKTRONIK 1973, H. 2, S. 47...52 und H. 3, S. 97...100.

Prof. Dr.-Ing. K. Hoffmann, Dipl.-Ing. H. Röder

Eine Meßdatenerfassungsanlage mit IEC-BUS

In der Meßpraxis nimmt der Einsatz der elektronischen Meßdatenerfassungs- und Verarbeitungsanlagen ständig zu. Hierbei bestimmt die jeweilige Meßaufgabe die Anzahl und die Art der elektronischen Meßgeräte, die zur Meßdatenerfassung zusammengeschaltet werden. Eine genormte Schnittstelle für die Verbindung von elektronischen Meßgeräten ist der IEC-Bus, der Elektronik-Lesern aus zahlreichen Aufsätzen bereits bekannt ist [1...6]. Im folgenden soll eine auf diesem Konzept realisierte Meßdatenerfassungsanlage beschrieben werden. Ausgehend von den Forderungen an das Meßdatenerfassungssystem werden die Schnittstellenfunktionen ausgewählt und die Zustandsdiagramme gezeigt. Danach folgt die Beschreibung des System-Steuergeräts, seine Programmstruktur und einige Programmierbeispiele.

1 Liste der Forderungen an das Meßdatenerfassungssystem

Um mit dem erstellten Meßdatenerfassungssystem die in der Praxis meist auftretenden Meßprobleme zu lösen, müssen folgende Forderungen erfüllt werden:

- Das System muß über ein Programmierfeld der jeweiligen Meßaufgabe angepaßt werden können.
- Das Meßprogramm muß in Einzelschritten überprüft werden können (Single-Step-Betrieb) und einfach, ohne Erlernen einer Programmsprache einstellbar sein.
- Das jeweilige Meßprogramm muß ohne Speicherabruf auf der Frontplatte des Systems erkennbar sein und nach einer Netzunterbrechung wieder verfügbar sein.
- Eine automatische periodische Meßdatenerfassung mit einstellbarer Meßperiode muß möglich sein.
- Eine Störungserkennung mit Lokalisierung des gestörten Gerätes soll automatisch erfolgen.
- Vorhandene Meß- und Datenspeicherungsgeräte ohne IEC-Bus-Schnittstelle sollen über einfache Interface-Schaltungen an das System angepaßt werden können. (Die Schnittstelle der Geräte muß hierzu bestimmten Mindestanforderungen genügen, die in den folgenden Ausführungen zusammengestellt werden.)

2 Auswahl der Schnittstellenfunktionen

Um die aufgestellten Forderungen zu erfüllen, müssen für die Interface-Schaltungen bestimmte Schnittstellenfunktionen aus der Norm ausgewählt werden. Die Auswahl der Schnittstellenfunktionen und die zugehörigen Zustandsdiagramme für das Meß- bzw. Datenspeicherungsgerät werden hier am Beispiel eines Digitalmultimeters und eines Lochstreifenstanzers beschrieben. Die im Text und in den Zustandsdiagrammen verwendeten Abkürzungen für Schnittstellenzustände und Nachrichten werden in *Tabelle 1* erläutert.

2.1 Die Schnittstellenfunktionen für ein Digitalmultimeter

Um ein Digitalmultimeter mit einer IEC-Bus-Schnittstelle versehen zu können, muß die gerätespezifische Schnittstelle mindestens folgende Eigenschaften aufweisen:

- Das Meßgerät muß einen Datenausgang besitzen, an dem Vorzeichen, Ziffern und Komma digital codiert zur Verfügung stehen.
- Es ist ein Triggereingang erforderlich, mit dem das Gerät zum Auslösen des Meßvorgangs und zur Speicherung des Meßwertes während der Datenübergabe veranlaßt werden kann.

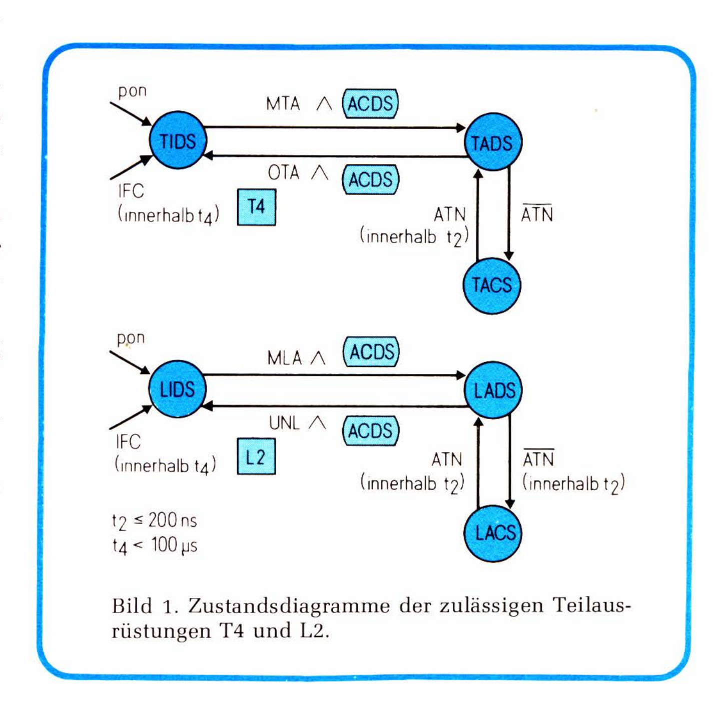


Bild 1. Zustandsdiagramme der zulässigen Teilausrüstungen T4 und L2.

Entsprechend diesen Forderungen ergibt sich unter den zehn genormten Schnittstellenfunktionen die minimale Auswahl, die in *Tabelle 2* angegeben ist.

Zur Realisierung der Talker- und Listener-Funktion genügen die in der Norm aufgeführten Teilausrüstungen T 4 und L 2, deren Zustandsdiagramme in *Bild 1* gezeigt werden. Da bei der T 4-Funktion die Zustände SPIS, SPMS und SPAS fehlen, reagiert sie nicht auf die Serien-Abfrage (Serial Poll). Dies ist unter zwei Voraussetzungen zulässig:

- Das Gerät selbst darf keinen Service Request senden.
- Die Talker-Funktion darf bei einer Serien-Abfrage nach einem Service Request durch ein anderes Gerät nicht aktiviert werden.

Da bei vielen Digitalmultimetern keine Möglichkeit besteht, den Meßbereich über Steuerleitungen extern einzustellen, kann die Remote/Local-Funktion entfallen. Dadurch wird es auch nutzlos, bei einer Meßbereichsüberschreitung einen Bedienungsruf (SRQ) zu senden, da die Systemsteuerung (Controller) darauf nicht wirkungsvoll reagieren kann. Somit braucht die Service Request-Funktion nicht realisiert zu werden.

2.2 Die Schnittstellenfunktionen für einen Lochstreifenstanzer

Bei der Auswahl der Schnittstellenfunktionen für die Interface-Schaltung zum Anschluß eines Lochstreifenstanzers wurde davon ausgegangen, daß die Stanzer-Schnittstelle folgende Anschlußleitungen besitzt:

- Eingangsleitungen zur Übernahme von Meßdaten auf den Lochstreifen und einen Triggereingang zum Auslösen des Stanzvorganges.
- Eine Ausgangsleitung für die Rückmeldung, daß der Stanzer bereit ist, neue Daten zu empfangen.
- Ausgangsleitungen für eine Fehlermeldung „Lochstreifen gerissen“ oder „nur noch wenige Meter Lochstreifen auf der Vorratsspule“. (Diese Ausgangsleitungen sind für den Anschluß eines Lochstreifenstanzers an den IEC-Bus nicht unbedingt erforderlich.)

Unter den genormten Schnittstellenfunktionen wurde für das Stanzer-Interface die in *Tabelle 3* gezeigte Auswahl getroffen.

Tabelle 1.
Abkürzungen für Schnittstellenzustände und Nachrichten

Schnittstellen-Zustände	
LIDS / TIDS	Ruhezustand des Hörers/Sprechers
LADS / TADS	Adressierter Zustand des Hörers/Sprechers
LACS / TACS	Aktiver Zustand des Hörers/Sprechers
der Steuerfunktion:	
CACS	Aktiver Zustand
CSBS	Bereitschaftszustand
CSWS	Synchronisier-Wartezustand
CAWS	Aktiver Wartezustand
CSNS	Bedienungsrufloser Zustand
CSRS	Bedienungsruf-Empfangszustand
der Systemsteuerung:	
SNAS	Inaktiver Zustand
SACS	Aktiver Zustand
SIIS	Rücksetz-Ruhezustand
SIAS	Aktiver Rücksetz-Zustand
SINS	Inaktiver Rücksetz-Zustand
SRIS	Fernsteuerfreigabe-Zustand
SRAS	Aktiver Fernsteuerfreigabe-Zustand
SRNS	Inaktiver Fernsteuerfreigabe-Zustand
der Quelle:	
SGNS	Erzeugungszustand
SDYS	Verzögerungszustand
STRS	Übertragungszustand
SWNS	Wartezustand
der Senke:	
ACRS	Bereitzustand
ACDS	Datenübernahmezustand
AWNS	Wartezustand
Externe Nachrichten:	
MTA	eigene Sprecheradresse
OTA	fremde Sprecheradresse
UNL	hören beenden
MLA	eigene Höreradresse
SRQ	Bedienungsruf
RFD	Bereit für Daten
DAV	Daten gültig
ATN	Achtung
GET	Gerätegruppe auslösen
REN	Fernsteuerung freigeben
Interne Nachrichten:	
pon	Gerät EIN
gts	in Bereitschaftszustand gehen
tcs	Steuerung synchron übernehmen
tca	Steuerung asynchron übernehmen
rpp	Parallelabfrage fordern
rsc	Systemsteuerung fordern
sic	Rücksetzbefehl (= IFC) senden
sre	Fernsteuerfreigabe (= REN) senden
nba	neues Byte verfügbar
ton	nur sprechen

Tabelle 2. Auswahl der Schnittstellenfunktionen für ein Digitalmultimeter

Schnittstellenfunktion	Ausführung
Talker-Funktion	T 4
Source-Handshake-Funktion	SH 1
Listener-Funktion	L 2
Acceptor-Handshake-Funktion	AH 1
Device-Trigger-Funktion	DT 1

Tabelle 3. Auswahl der Schnittstellenfunktionen für einen Lochstreifenstanzer

Schnittstellenfunktion	Ausführung
Talker-Funktion	T 6
Source-Handshake-Funktion	SH 1
Listener-Funktion	L 4
Acceptor-Handshake-Funktion	AH 1
Service-Request-Funktion	SR 1

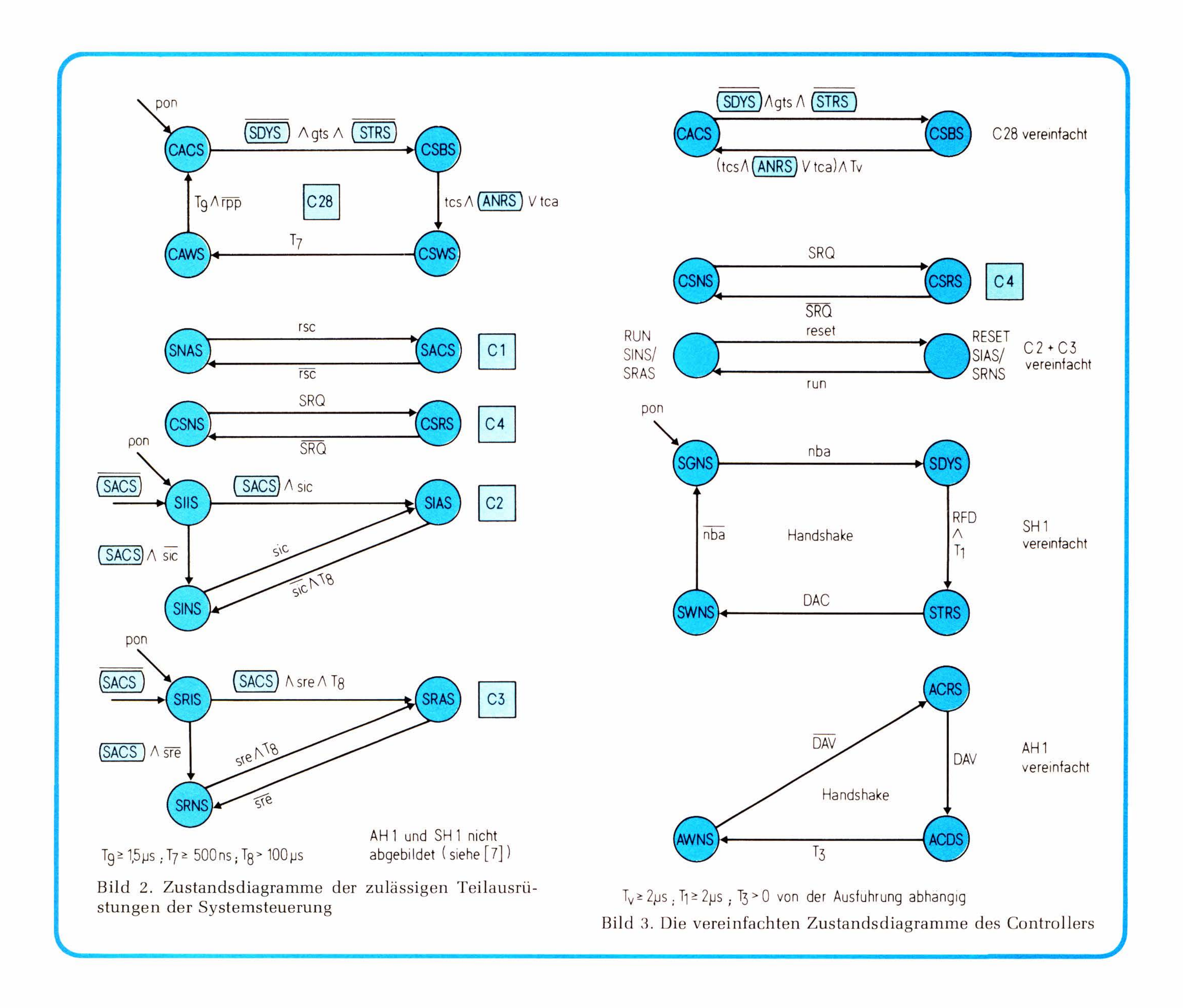


Bild 2. Zustandsdiagramme der zulässigen Teilausrüstungen der Systemsteuerung

Bild 3. Die vereinfachten Zustandsdiagramme des Controllers

2.3 Die Zustandsdiagramme der Systemsteuerung

Zur Erfüllung der geforderten Fähigkeiten an das Meßdatenerfassungssystem muß die Systemsteuerung (Controller) die in *Tabelle 4* gezeigten Schnittstellenfunktionen besitzen. Die Zustandsdiagramme dieser zulässigen Teilausrüstungen des Controllers werden in *Bild 2* gezeigt.

Bei der Auswahl der Schnittstellenfunktionen für den Controller wurde auf die Fähigkeit verzichtet, die Steuerung an ein anderes Systemsteuergerät zu übergeben oder von diesem zu übernehmen. Die Ermittlung der Adresse eines SRQ sendenden Gerätes wird mit der Serienabfrage durchgeführt, wodurch die Parallelabfrage entfallen konnte.

Ausgehend von den genormten Zustandsdiagrammen des Controllers kann man verschiedene Vereinfachungen vornehmen, die zu den Zustandsdiagrammen in *Bild 3* führen. Bei den Vereinfachungen wurde darauf geachtet, daß der Controller die geforderten Fähigkeiten erfüllt und der Aufwand zur Realisierung der Controllerschaltung wesentlich reduziert wird.

In Bild 2 wurden die Zustände CIDS, CADS, CTRS, die zur Übergabe der Steuerung dienen, weggelassen. Ebenfalls entfallen die Zustände CPWS und CPPS, da keine Parallelabfrage durchgeführt werden soll. Daraus ergibt sich außerdem rpp=o, wodurch der Übergang vom Zustand CSBS nach CACS nur noch eine Zeitbedingung darstellt (Bild 3).

Die zulässige Teilausrüstung C 1 der Controllerfunktion entfällt, da der Controller sich immer in einem aktiven Zustand befinden soll und kein anderer Controller zur Systemsteuerung zugelassen wird.

Eine weitere Vereinfachung stellt die Zusammenfassung der Controllerfunktionen C 2 und C 3 dar. Hierdurch werden zwei Betriebszustände geschaffen:

Tabelle 4. Auswahl der Schnittstellenfunktionen für die Systemsteuerung

Schnittstellenfunktion	Ausführung
Source-Handshake-Funktion	SH 1
Acceptor-Handshake-Funktion	AH 1
C-Schnittstellenfunktion:	
Systemsteuerung	C 1
IFC senden und in Einsatz gehen	C 2
REN senden	C 3
auf SRQ ansprechen	C 4
Schnittstellennachrichten senden	C 28

- RUN: IFC = F, REN = T
  Es sind Bus-Aktivitäten (Daten/Adressenübertragung) möglich; die angeschlossenen Geräte müssen auf Bus-Befehle reagieren.
- RESET: IFC = T, REN = F
  Es sind keine Bus-Aktivitäten möglich; die Geräte brauchen nicht auf Bus-Befehle zu reagieren.

Der Controller kann mit einer RESET/RUN-Taste an der Gerätefrontplatte in beide Betriebszustände geschaltet werden.

Im RUN-Zustand des Controllers sind zwei verschiedene Controller-Aktivitäten möglich:

- Ablauf des eingestellten Meßprogramms.
- Ermittlung der Adresse eines SRQ sendenden Gerätes mittels Serial Poll.

Hierzu benötigt der Controller noch die AH- und die SH-Funktion. Diese Funktionen wurden ebenfalls vereinfacht, indem Zustände, die aufgrund des vorliegenden Konzepts nie eingenommen werden, weggelassen wurden. Dies sind insbesondere die Zustände, bei denen das Übergangskriterium zu ihrem Verlassen immer erfüllt ist.

Es muß an dieser Stelle darauf hingewiesen werden, daß diese Vereinfachungen der Zustandsdiagramme in der Norm nicht vorgesehen sind [7]. Die genormten Zustandsdiagramme werden in Bild 2 gezeigt.

3 Programmstruktur und Programmierung des Meßdatensystems

Um verschiedenartige Meßprobleme lösen zu können, wird der Controller über ein Programmierfeld auf das jeweilige Meßprogramm eingestellt. Als Programmierfeld wird ein Kreuzschienenverteiler an der Gerätefrontplatte verwendet. Das Meßprogramm wird über Stifte, die in den Kreuzschienenverteiler gesteckt werden, eingegeben und ist jederzeit ohne Erlernen einer Programmsprache einstellbar.

Die Programmstruktur des Controllers besteht aus einem festverdrahteten Hauptprogrammteil und einem Unterprogrammteil, der über das Programmierfeld eingegeben wird. Im Hauptprogramm sind immer wiederkehrende Befehle der Meßprogramme fest verdrahtet. An bestimmten Stellen im Hauptprogramm erfolgt ein Sprung in das Unterprogramm. Vom Unterprogramm aus kann an frei wählbarer Stelle die Rückkehr in das Hauptprogramm eingegeben werden (return-Stift). Der grundsätzliche Programmablauf wird in *Bild 4* gezeigt.

Das Unterprogramm enthält im allgemeinen programmierte Adressen und Befehle. Im Hauptprogramm werden die Befehle

UNL ...	hören beenden
GET ...	Gerätegruppe auslösen
und UNT ...	sprechen beenden

gesendet.

Da bei dem IEC-Bus-System nur ein Talker auf dem Bus aktiv sein darf, wird bei der Aussendung einer neuen Talker-Adresse durch den Controller der alte Talker automatisch deadressiert. Der hier verwendete Befehl UNT (Kodierung 1011111) ist in der IEC-Bus-Norm nicht explizit vorgesehen. Ein Aussenden der oben gezeigten Codierung bewirkt jedoch auf dem IEC-Bus eine Deadressierung aller Talker, da die Talker-Adresse 11111 verboten ist.

In *Bild 5* wird anhand eines einfachen Beispiels die Programmierung des Meßdatenerfassungssystems gezeigt. Ein als Hörer adressiertes Meßgerät wird durch den Befehl GET zur Auslösung des Meßvorgangs und zur Speicherung des Meßwertes veranlaßt. Danach wird ein als Hörer adressiertes Datenspeichergerät den Meßwert des jetzt als Sprecher adressierten Meßgerätes übernehmen. Für diese Meßaufgabe werden drei Programmspalten des Kreuzschienenverteilers benötigt (*Bild 6*). Bei dem hier verwendeten Kreuzschienenverteiler mit 10 Programmspalten kann man noch zusätzlich zur Kennzeichnung der vom Meßgerät gesendeten Daten sieben Zeichen (Bytes) programmieren.

Ein anderes Meßprogramm, das die Speicherung der Meßwerte von vier verschiedenen Meßgeräten mit einem Datenspeichergerät (z. B. Lochstreifenstanzer) bewirkt, wird in *Bild 7* gezeigt.

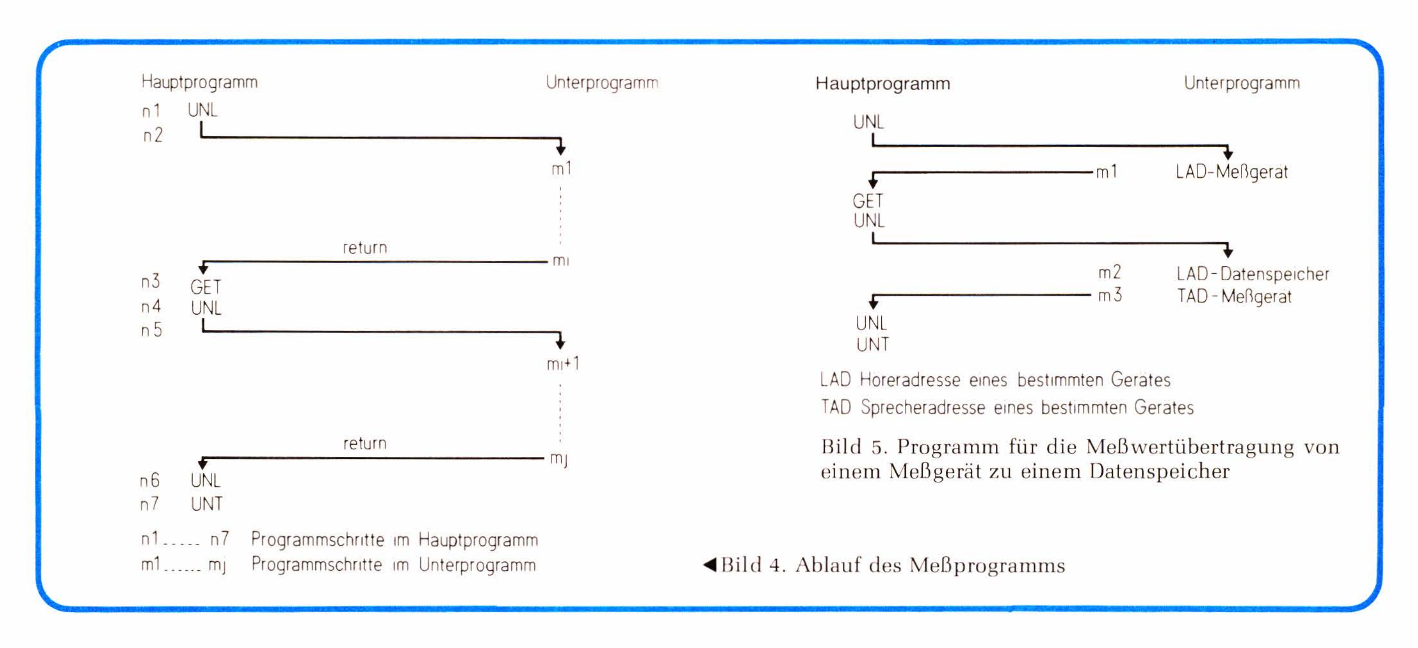


Bild 5. Programm für die Meßwertübertragung von einem Meßgerät zu einem Datenspeicher

◀Bild 4. Ablauf des Meßprogramms

Alle Programme können mit Hilfe einer Single-Step Taste in Einzelschritten überprüft werden und sind auf dem Kreuzschienenverteiler durch farbige Stifte erkennbar. Ein eingebauter Zeitgeber erlaubt die automatische periodische Meßdatenerfassung mit einstellbarer Meßperiode im Bereich von 1 s...99 min. Im Störungsfall wird die Adresse eines SRQ sendenden gestörten Gerätes gespeichert und auf der Frontplatte angezeigt.

Bild 8 zeigt das Blockschaltbild der erstellten Meßdatenerfassungsanlage.

4 Aufbau der Systemsteuerung

Damit der Controller die angegebenen Aufgaben erfüllen kann, besitzt er vier Grundzustände (Modes), die in *Bild 9* in ihrem logischen Zusammenhang dargestellt sind.

Die Bedeutung der Zustände:

- SPM: Serial Poll Mode. In diesem Zustand wird das Service Request sendende Gerät durch eine Serienabfrage ermittelt. Dieser Zustand wird durch SRQ = Low (true) erreicht, wenn diese Betriebsart durch einen Schalter „SP" gewählt wurde.
- PRM: Program Mode. In diesem Zustand läuft das im Programmierfeld eingesteckte Meßprogramm ab. Der Zustand wird manuell durch die Taste „Start Programm" oder automatisch durch ein Signal des Zeitgebers erreicht.

In den Zuständen SPM und PRM kann der Controller wahlweise in Einzelschrittsteuerung oder automatisch betrieben werden. Die Bedeutung der Zustände RES und RUN wurde in Abschnitt 2.3 erläutert.

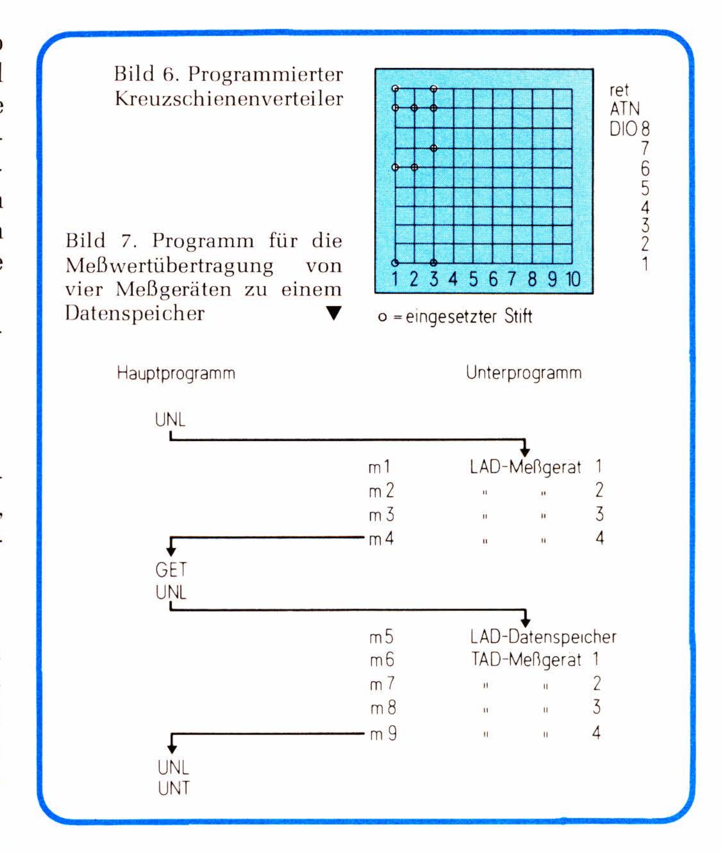
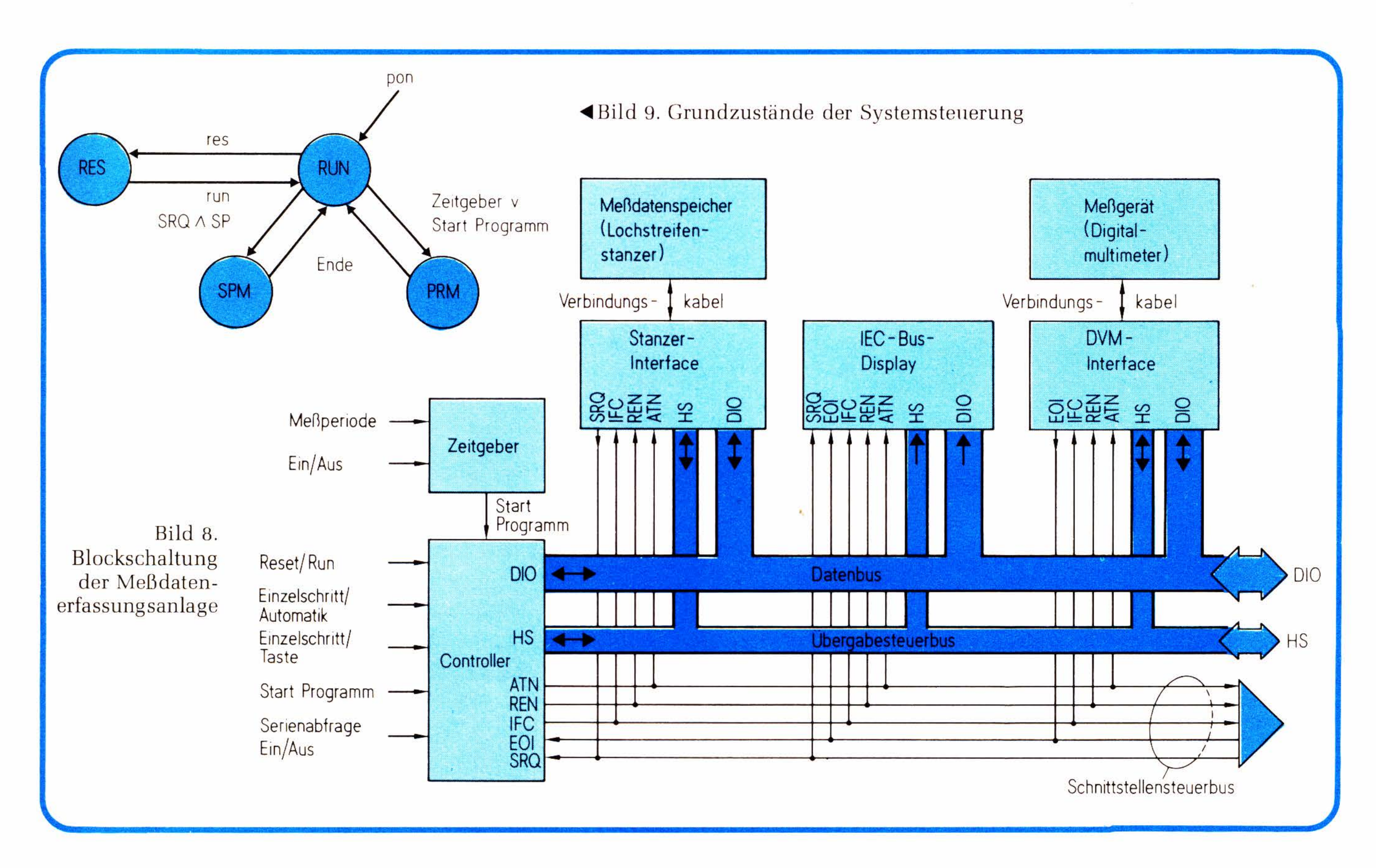


Bild 6. Programmierter Kreuzschienenverteiler

Bild 7. Programm für die Meßwertübertragung von vier Meßgeräten zu einem Datenspeicher ▼

Das Blockschaltbild des Controllers ist in *Bild 10* gezeigt. Man erkennt, daß zu jedem Zustand aus Bild 9 eine Baugruppe gehört. Die hier vorgenommene Aufteilung des Controllers in teilbetriebsfähige Baugruppen reduziert die Anzahl der Verbindungsleitungen und ermöglicht einen einfachen Nachweis zur Einhaltung der Spezifikationen.



◀Bild 9. Grundzustände der Systemsteuerung

Bild 8. Blockschaltung der Meßdatenerfassungsanlage

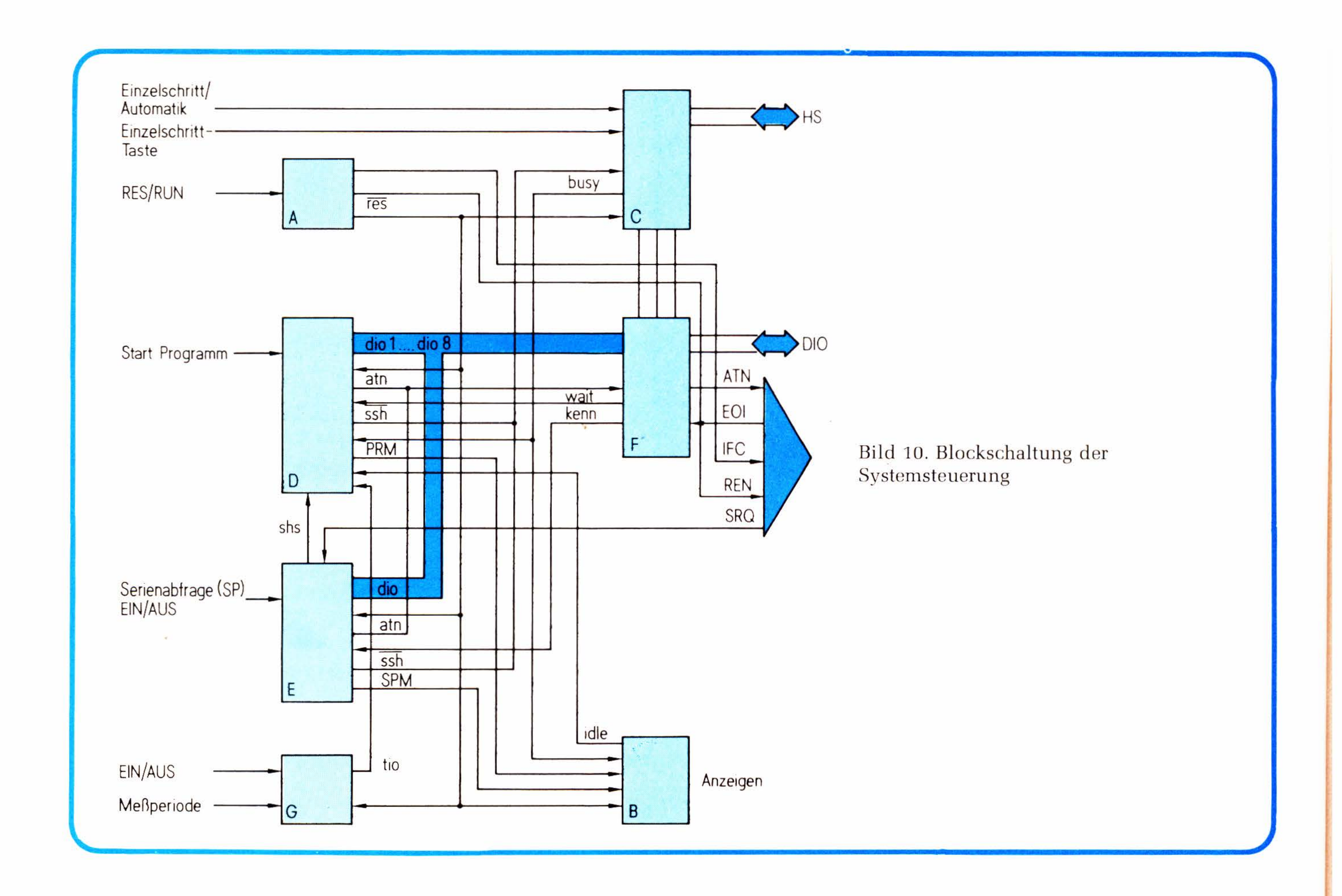


Bild 10. Blockschaltung der Systemsteuerung

4.1 Die Funktion der Baugruppen

Die Funktion der Baugruppen des Controllers soll im folgenden unter Verwendung von Bild 10 erklärt werden.

Die Baugruppe A steuert die IEC-Bus-Leitungen IFC und REN. Außerdem liefert sie das Reset-Signal für alle Speicherelemente des Controllers ($\overline{res}$).

Die Baugruppe E erzeugt den Ablauf der Serienabfrage (Serial Poll). Die hierzu benötigten genormten Befehle, Sprechadressen und ein Signal zum Starten des Source-Handshake ($\overline{shs}$) werden erzeugt. Die Kennung (Identifizierung eines Gerätes, das einen Bedienungsruf „SRQ" gesendet hat) wird im Acceptor-Handshake empfangen (Baugruppe C), ausgewertet (Baugruppe F) und der Baugruppe E zugeführt (kenn).

Die Baugruppe D erzeugt Ablauf und Befehle des Programms. Hierzu wird ebenfalls ein Signal zum Starten des Source-Handshake erzeugt ($\overline{ssh}$). Vom „Ende-Decoder" (Baugruppe F) erhält die Baugruppe D das Signal $\overline{wait}$, wenn ein adressierter Sprecher seine Datenübertragung beendet hat.

Die Baugruppen D und E erzeugen somit zusammen das atn-Signal.

Die Baugruppe C enthält die Schaltungen für Source- und Acceptor-Handshake, die Einzelschrittsteuerung (Single-Step) und den automatischen Betrieb. Das in dieser Baugruppe erzeugte busy-Signal steuert als Takt den Programmablauf für das Meßprogramm oder die Serienabfrage. Dieses Signal ist nur von der Dauer des Source-Handshake auf dem IEC-Bus abhängig, so daß die Geschwindigkeit des Programmablaufs im wesentlichen nur von den am IEC-Bus angeschlossenen Geräten bestimmt wird.

In der Baugruppe B werden die Grundzustände des Controllers decodiert und an der Gerätefrontplatte zur Anzeige gebracht. Das Signal idle (Ruhezustand der Systemsteuerung) wird aus den Zuständen des Controllers in der Weise idle = $\overline{RES} \wedge \overline{SPM} \wedge \overline{PRM} \wedge \overline{busy}$ decodiert.

Der Zeitgeber in Baugruppe G liefert das Signal zum Starten des Meßprogramms für die automatische periodische Meßdatenerfassung (tio).

4.2 Die Programmsteuerung

In den folgenden Ausführungen soll nun die Hardware der Programmsteuerung näher erklärt werden. Durch die Auswahl der Schnittstellenfunktionen und durch die Angabe der betreffenden Zustandsdiagramme ist auch die Realisierung der restlichen Baugruppen möglich [8, 6]. Schaltungen zur Erzeugung dieser Schnittstellenfunktionen befinden sich auch in den Aufsätzen [6, 1].

Das Blockschaltbild der Programmsteuerung zeigt *Bild 11*.

Es sei davon ausgegangen, daß die Schieberegister SR 1 (Steuerung Hauptprogramm) und SR 2 (Steuerung Unterprogramm) zurückgesetzt sind, d. h. alle Ausgänge n 1...n 8 und m 1...m 10 besitzen den Signalpegel „Low". In diesem Zustand wird an den Dateneingängen A ein High-Signal erzeugt und der Taktauswahlschalter steht in der Stellung a. Der erste Taktimpuls schiebt das High-Signal vom Dateneingang A

in die erste Registerzelle des Schieberegisters SR 1 (d. h. n 1 = High). Dadurch verschwindet das High-Signal an A von SR 1 und das High-Signal am Ausgang n 1 des Schieberegisters bewirkt, daß das erste Befehlswort (UNL) auf den internen Datenbus geschaltet wird. An diesen internen Bus sind die Schaltungen mit den festverdrahteten Befehlen und der Kreuzschienenverteiler angeschlossen. Der interne Bus ist in Bild 11 mit dio 1...dio 8 bezeichnet.

Der erste Taktimpuls bewirkt noch, daß ein Signal zum Starten eines Source-Handshake erzeugt, und daß auf einer internen Leitung das Signal atn aktiv wird (dieses Signal bewirkt, daß auf dem IEC-Bus ATN aktiv Low wird). Hierdurch wird erreicht, daß der erste Befehl (hier UNL) über den IEC-Bus gesendet wird.

Der zweite Taktimpuls schiebt das High-Signal im Schieberegister von n 1 nach n 2. Jetzt wird der Takt-Auswahlschalter nach b gestellt, und das Schieberegister SR 2 (Steuerung-Unterprogramm) erhält einen Taktimpuls. Das High-Signal am Eingang A von Schieberegister SR 2 wird nach m 1 geschoben. Dadurch verschwindet das High-Signal an A von SR 2, das Programmierfeld wird aktiviert und das erste Datenwort im Unterprogramm (PW 1) auf den dio-Bus geschaltet. Danach erfolgt die Übertragung des programmierten Datenwortes PW 1 durch Starten eines Source-Handshake und Erzeugung des Signals ATN auf dem IEC-Bus. Die folgenden Taktimpulse schieben das High-Signal durch das Schieberegister SR 2 und bewirken die Übertragung der im Programmierfeld eingesteckten Datenwörter, und zwar so lange, bis

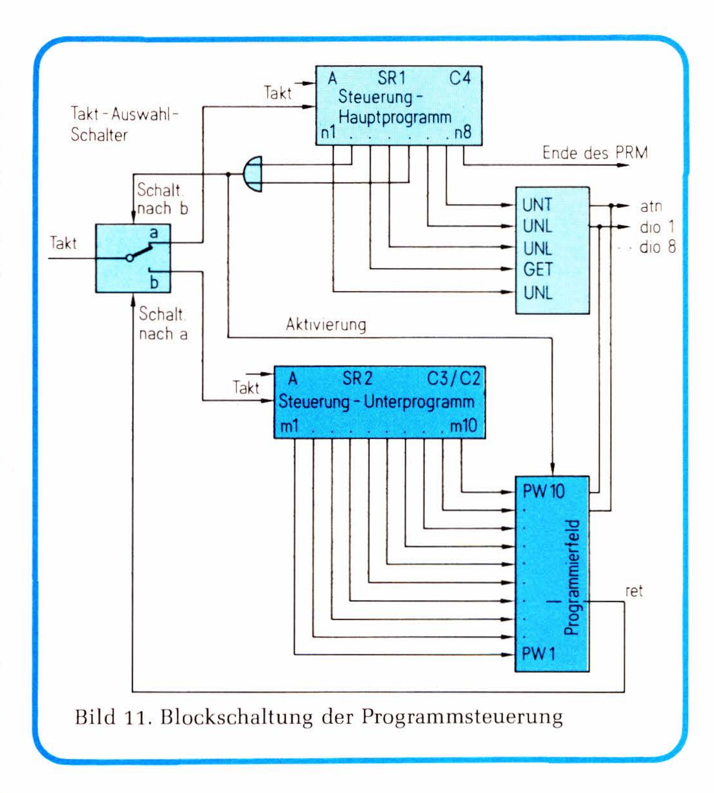


Bild 11. Blockschaltung der Programmsteuerung

Bild 12. Schaltung der Programmsteuerung

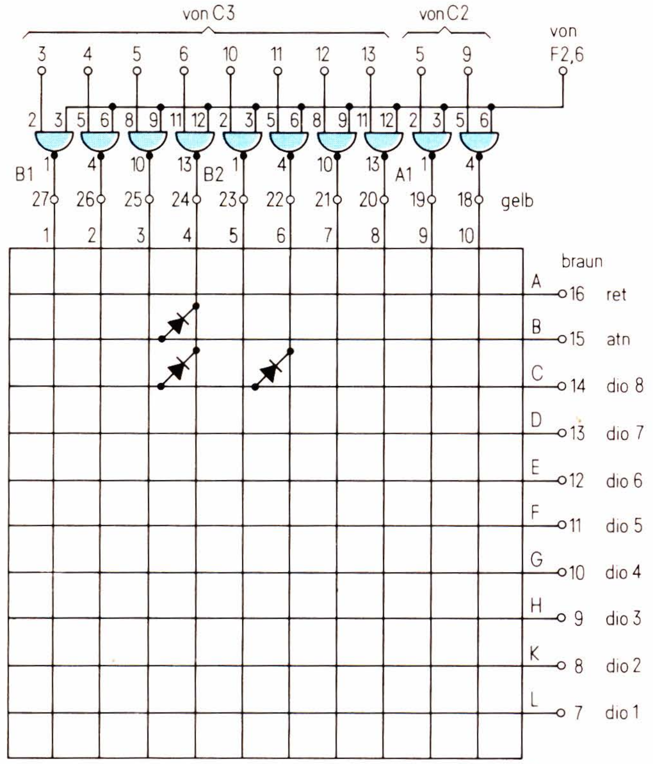


Bild 13. Detailschaltung Kreuzschienenverteiler (= Programmierfeld)

dort die Information ret (= Rückkehr ins Hauptprogramm) gesetzt ist. Das Signal ret schaltet den Taktauswahlschalter nach a. (Beispiel Bild 11: ret ist im Schritt m 4 gesetzt. Nach Übertragung des dort programmierten Datenwortes PW 4 erfolgt die Umschaltung des Takts.)

Der nächste Taktimpuls wird nun wieder im Schieberegister SR 1 wirksam und schiebt das High-Signal nach n 3. Nach der Übertragung von GET und UNL über den IEC-Bus erfolgt bei n 5 = High nochmals ein Sprung ins Unterprogramm (zum oben genannten Beispiel: Sprung nach m 5). Die Rückkehr ins Hauptprogramm erfolgt, wie oben beschrieben, über die Programmierung von ret. Nach der Übertragung der Nachrichten UNL und UNT gelangt das High-Signal im Schieberegister SR 1 nach n 8 und beide Schieberegister (SR 1, SR 2) werden gelöscht. Damit ist ein Programmzyklus beendet. Dieser Programmzyklus kann mit dem einstellbaren Zeitgeber periodisch wiederholt werden.

4.3 *Dokumentation der Programmsteuerungen*

Die Dokumentation und den Aufbau der beschriebenen Programmsteuerung dieses Meßdatenerfassungssystems zeigen die *Bilder 12* und *13*. Als Bauelementbezeichnung ist die Platinenposition der Bauelemente gewählt.

Bild 14 zeigt ein Foto der in 10 Mannmonaten erstellten Meßdatenerfassungsanlage.

Die in dem hier vorgestellten System verwendeten Baugruppen wurden von den Herren *Dipl.-Ing. W. Handke*, *Dipl.-Ing. G. Wischermann* und *Dipl.-Ing. H. M. Gößwein* im Rahmen ihrer Diplomarbeit erstellt und getestet.

Literatur

[1] *Pannach, A.*: Interface-Entwicklung für den IEC-Bus. ELEKTRONIK 1975, H. 12, S. 61...64.

[2] *Klaus, J.*: Wie funktioniert der IEC-Bus? ELEKTRONIK 1975, H. 4, S. 72...78 und H. 5, S. 73...78.

[3] *Freytag, H. H.*: Ein Sieg der Vernunft. ELEKTRONIK 1975, H. 1, S. 3.

[4] *Schmidt, U.*: Stand der Arbeiten am IEC-Bus-System. ELEKTRONIK 1975, H. 1, S. 58.

[5] *Richert, U.*: IEC-Bus-Interface für Prozeßrechner. ELEKTRONIK 1976, H. 12, S. 58...62.

[6] *Richter, M.*: Das Zustandsdiagramm und seine Anwendung beim IEC-Bus. ELEKTRONIK 1977, H. 2, S. 55...58 und S. 71.

[7] Entwurf DIN IEC 66.22.

[8] *Künzel, R.*: Das State-Diagramm. ELEKTRONIK 1973, H. 2, S. 47...52 und H. 3, S. 97...100.

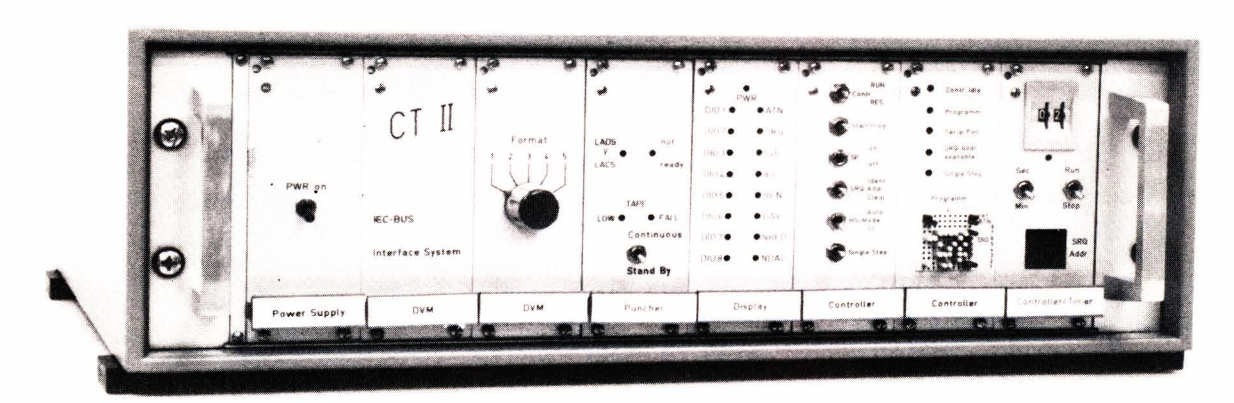


Bild 14. Praktische Realisierung der Meßdatenerfassungsanlage

Ing. (grad.) Georg Unger

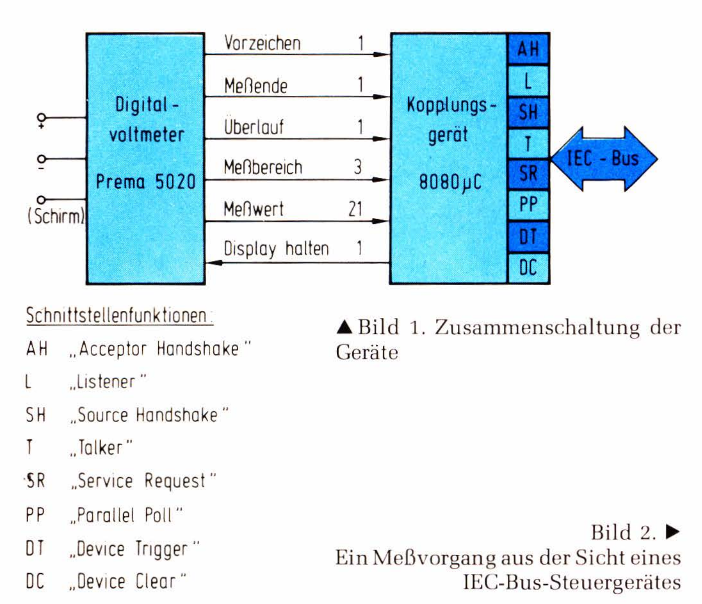
IEC-Bus-Kopplungsgerät für Digitalvoltmeter

Der Entwurf einer Schnittstelleneinheit zwischen Digitalvoltmeter und IEC-Bus – im folgenden Kopplungsgerät genannt – ist damit zu begründen, daß in vielen Labors eine Reihe leistungsfähiger Digitalmeßgeräte mit parallelem Datenausgang zur Verfügung stehen, die keine IEC-Bus-Schnittstelle besitzen. Im folgenden Aufsatz wird beschrieben, wie ein Kopplungsgerät aussehen muß, damit es mit einem Digitalvoltmeter zu einem selbständigen IEC-Bus-kompatiblen Einzelgerät wird.

1 Zielsetzung

Heute werden von vielen Meßgeräteherstellern IEC-Bus-kompatible Digitalvoltmeter, insbesondere Digitalmultimeter angeboten, die speziell für den Einsatz in IEC-Bus-Systemen [1] konzipiert wurden. Kritisch sei vermerkt, daß viele dieser Meßgeräte für leistungsfähige IEC-Bus-Systeme ungeeignet sind, da sie dem Steuergerät wenig oder keine Verarbeitungsroutinen abnehmen. Wann z. B. nach einem Meßauftrag ein gültiger Meßwert zur Weiterverarbeitung übertragen werden kann, muß das Steuergerät entscheiden; dadurch ist es für die Zeit des Gerätezyklus damit beschäftigt, auf Meßwerte zu warten, diese abzuholen und zu verarbeiten.

Ein wirklich „Bus-taugliches" Gerät kann darüber hinaus eigene Entscheidungen treffen, Fehler erkennen und sich aktiv beim Steuergerät melden. Für den Einsatz in IEC-Bus-Systemen mit „verteilter Intelligenz" muß an ein Einzelgerät diese Forderung gestellt werden.



▲ Bild 1. Zusammenschaltung der Geräte

2 Das Kopplungsgerät als IEC-Bus-kompatible Einheit

Zum Anschluß an den IEC-Bus wurde in vorliegendem Fall das Digitalvoltmeter 5020 (Prema) [2] mit einem Kopplungsgerät ausgestattet (*Bild 1*). Es bildet die Schnittstelle zwischen Digitalvoltmeter einerseits und dem IEC-Bus andererseits.

In *Bild 2* sind die Aktivitäten zur Ansteuerung des Kopplungsgeräts über den IEC-Bus aus der Sicht eines Steuergerätes dargestellt:

Nachdem das Kopplungsgerät als Hörer adressiert wurde, wird durch den „adressierten" Befehl GET (*Group Execute Trigger*) ein Gerätezyklus ausgelöst. Ab diesem Zeitpunkt arbeitet das Kopplungsgerät unabhängig vom IEC-Bus. Dadurch werden IEC-Bus-Systeme, die während der busunabhängigen Aktivitäten von Einzelgeräten den Bus für andere Aufgaben nutzen, entlastet [3]. Das Kopplungsgerät kommuniziert mit dem Digitalvoltmeter und erarbeitet Meßwert, Begleitzeichen und Statusinformation. Nach Erledigung meldet es sich aktiv über die SRQ-Busleitung beim Steuergerät. Das Kopplungsgerät kann sofort oder zu einem späteren Zeitpunkt als Sprecher adressiert

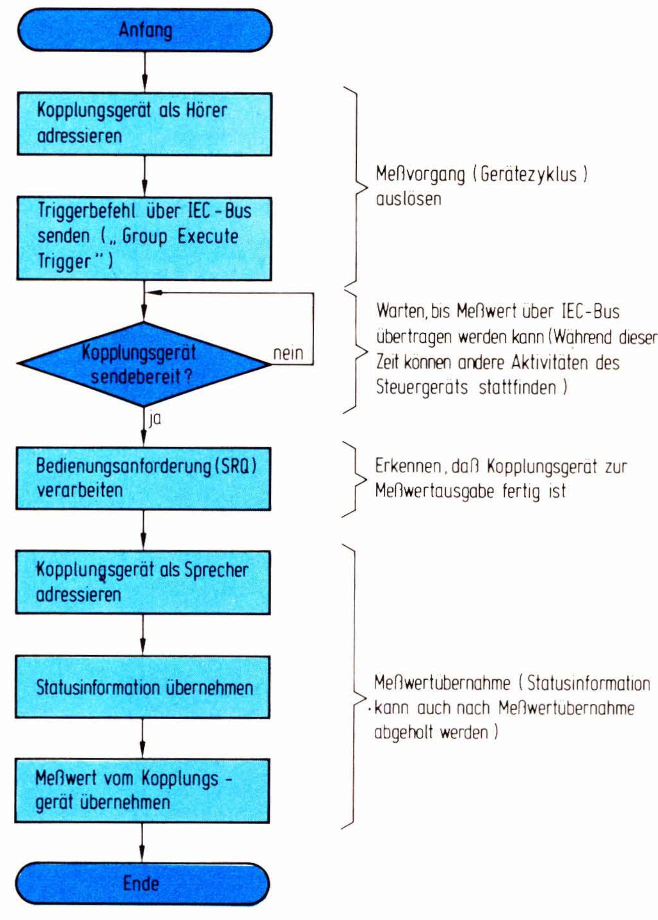


Bild 2. ▶
Ein Meßvorgang aus der Sicht eines IEC-Bus-Steuergerätes

werden, um einen Meßwert oder die Statusinformation zu senden. Wenn das Steuergerät das dem Meßwert zugeordnete Statusbyte übernommen hat, nimmt das Kopplungsgerät die Bedienungsanforderung zurück.

Das Kopplungsgerät ist mit einem IEC-Bus-LSI-Schnittstellenbaustein HEF 4738 V von Valvo ausgestattet [4]. Die in Bild 1 aufgeführten Schnittstellenfunktionen wurden verwendet. Die Gerätefunktion RL („Remote/Local“) ist nicht eingesetzt, da das angeschlossene Digitalvoltmeter keine Frontplattenabschaltung hat. Es ist sinnvoll, das Digitalvoltmeter mit automatischer Bereichswahl zu betreiben, um seinen Bedienungskomfort auszunutzen.

3 Hardwareaufbau des Kopplungsgeräts

Eine Übersicht der Schaltung ist in *Bild 3* als Blockbild dargestellt. Zusammen mit den *Bildern 4* und *5* ist die komplette Schaltung des Geräts wiedergegeben; auf die übliche Zusammenschaltung von 8080 – CPU mit den Bausteinen 8224 und 8228 wurde in der Darstellung verzichtet [5].

Die Schaltung enthält zwei programmierbare periphere Interface-Bausteine (PPI) vom Typ 8255. Einer ist für die Betriebsart 0, d. h. einfache Ein-Ausgabe, programmiert und hat die Aufgabe, den Datenausgang des Digitalvoltmeters zu bedienen (Bild 3 und 5). Kanal A, B und die niederwertigen 4 Bit des Kanals C sind Eingänge, die höherwertigen 4 Bit Ausgänge. Die an den beiden Datenselektoren (74LS157) anstehende 16-Bit-Meßinformation vom Digitalvoltmeter wird durch zwei Input-Befehle eingeholt.

Das Anzeigestop-FF wird durch das µP-Programm gesteuert und hat die Aufgabe, das Display des Digitalvoltmeters ab Meßwertbereitstellung nach einem Triggerbefehl bis zur vollständigen Ausgabe über den IEC-Bus festzuhalten. Dadurch wird der zu übergebende Meßwert angezeigt, auch wenn sich die Meßspannung ändert.

Das Meßende-FF wird durch den entsprechenden Digitalvoltmeter-Ausgang gesetzt und meldet die Meßdaten „gültig“. Der Zustand des Flipflops wird durch das µP-Programm abgefragt und zurückgesetzt.

Der zweite PPI ist für die Betriebsart 1, d. h. Ein-Ausgabe mit Quittierungssignalen, programmiert. Kanal A ist der Meßdatenausgang, Kanal B der Statusbyteausgang (Bild 3). Sobald ein Meßdatum im Ausgaberegister des Kanals A bereitsteht, wird dies durch einen Impuls am $\overline{OBF}_A$-Ausgang („Datumstrobe“) quittiert und das Daten-FF gesetzt (Bild 4). Dem IEC-Schnittstellenbaustein HEF 4738 V wird dies durch das Zweidrahthandshakesignal nba (*new byte available*) gemeldet. Die Rückmeldung erfolgt durch dcd (*don't change data*) aus der SH-Funktion; nba wird daraufhin zurückgenommen und das Quittungssignal auf den $\overline{ACK}_A$-Eingang gegeben.

Weiter wird durch dcd das Datenübernahme-FF gesetzt, das durch das µP-Programm bedient wird. Das nächste Datum eines Meßwertes wird durch einen Output-Befehl in das Ausgaberegister des Datenka-

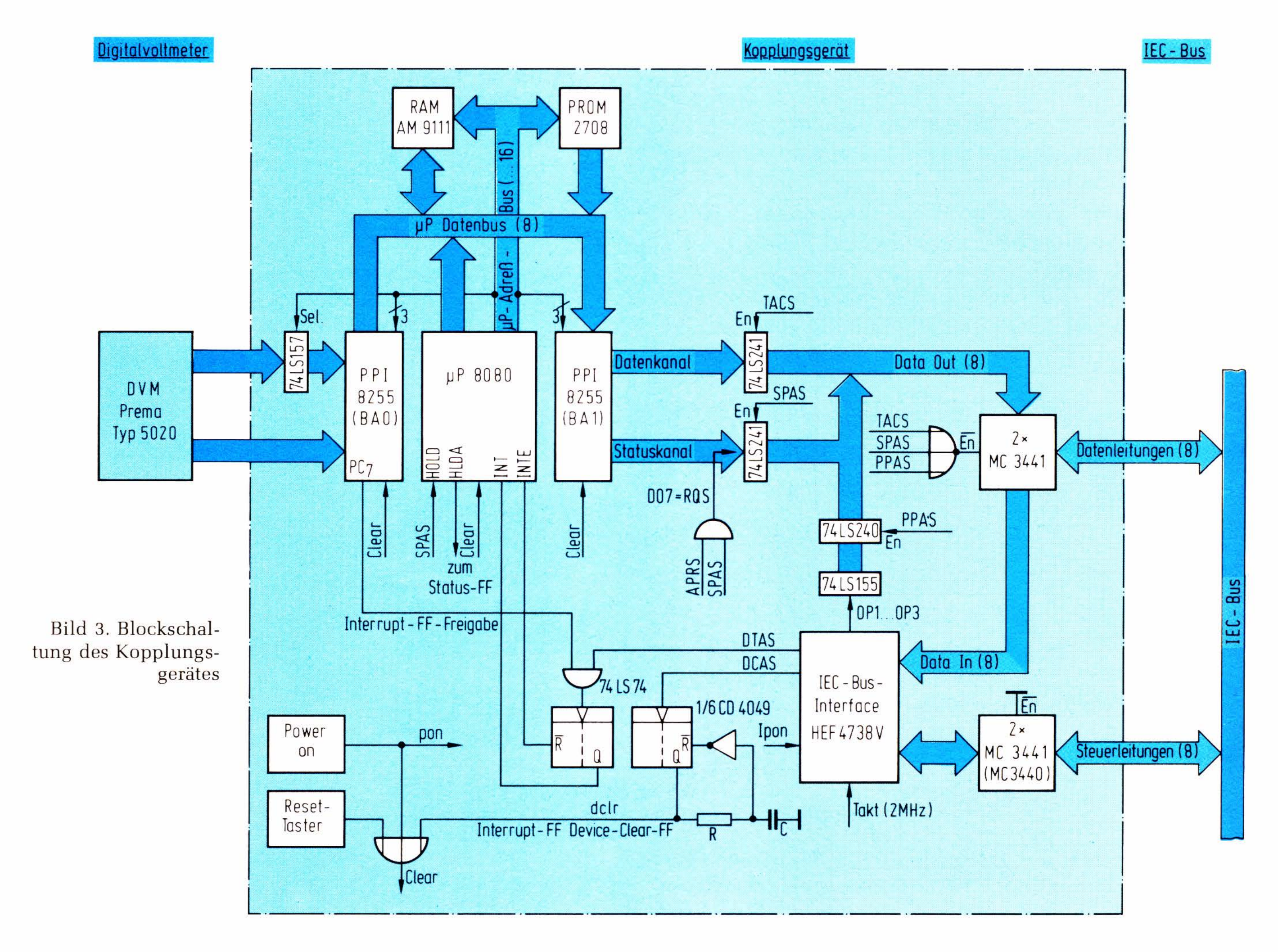


Bild 3. Blockschaltung des Kopplungsgerätes

nals geschrieben. Jedem auszugebenden Meßwert ist ein Statusbyte zugeordnet, das im Ausgaberegister des Kanals B bereitgestellt wird und über den $\overline{OBF}_B$-Ausgang („Statusstrobe") signalisiert wird. Dadurch wird das Bedienungsanforderungs-FF gesetzt und Signal rsv/ist („request service"/„individual status") geht an die IEC-Schnittstellenfunktionen SR und PP. Die IEC-Busleitung SRQ wird aktiviert, und die Statusby-

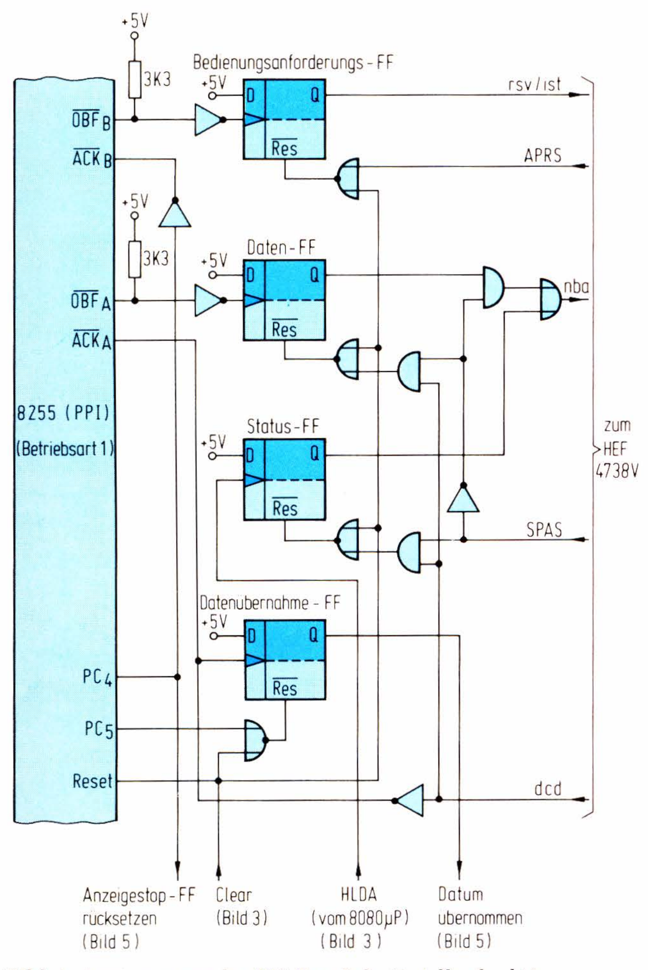


Bild 4. Ansteuerung der IEC-Bus-Schnittstellenfunktionen

teübertragung kann durch das Steuergerät eingeleitet werden. Während der Übertragung ist die Talker-Funktion im Zustand SPAS. Das dazugehörige Ausgangssignal gelangt an den HOLD-Eingang des 8080; die CPU schaltet sich nach Ende des momentanen Maschinenzyklus von den Bussen ab und sendet das Quittungssignal HLDA (Bild 3). HLDA setzt das Status-FF und damit nba; der Handshake zur Übertragung des Statusbyte ist eingeleitet. Nach der seriellen Abfrage verschwindet SPAS, und der Prozessor wird wieder aktiv.

Wird durch das Steuergerät eine Parallelabfrage durchgeführt, reagiert der HEF 4738 V durch eine der Sekundärnachricht PPE (Parallel Poll Enable) entsprechende Bitkombination an den Ausgängen OP1, OP2 und OP3. Die Signale sind auf einen Demultiplexer (3 zu 8) vom Typ 74LS155 geführt, an dessen Ausgang die 8-Bit-Parallel-Poll-Antwort über Tri-State-Treiber auf den IEC-Bus im Zustand PPAS gelangt (Bild 3). Die Übernahme erfolgt ohne Sprecheradressierung des Kopplungsgeräts und ohne Handshake-Abwicklung.

Die für die Schaltung benötigte Rückmeldeinformation wird aus dem Schnittstellenbaustein HEF 4738 V erzeugt:

TACS (*Talker Active State*) $\triangleq \overline{IATN} \wedge Ota \wedge \overline{Osp}$
SPAS (*Serial Poll Active State*) $\triangleq \overline{IATN} \wedge Ota \wedge Osp$
PPAS (*Parallel Poll Active State*) $\triangleq \overline{OPP}$
APRS (*Affirmative Poll Response State*) $\triangleq \overline{ORQS}$

Ein Gerätezyklus wird durch GET ausgelöst, der den Otrg-Ausgang im Zustand DTAS aktiviert und das Interrupt-FF setzt (Bild 3). Das Interrupt-Signal ist mit dem INT-Eingang des 8080 verbunden, die CPU verläßt den Halt-Zustand und erzeugt die Restart-7-Adresse. Die Erarbeitung des Meßwertes (Gerätezyklus) läuft als Interrupt-Service-Routine ab. Das Setzen des Interrupt-FFs ist während dieser Zeit durch das Signal „INT-FF-Freigabe" verriegelt, damit eine

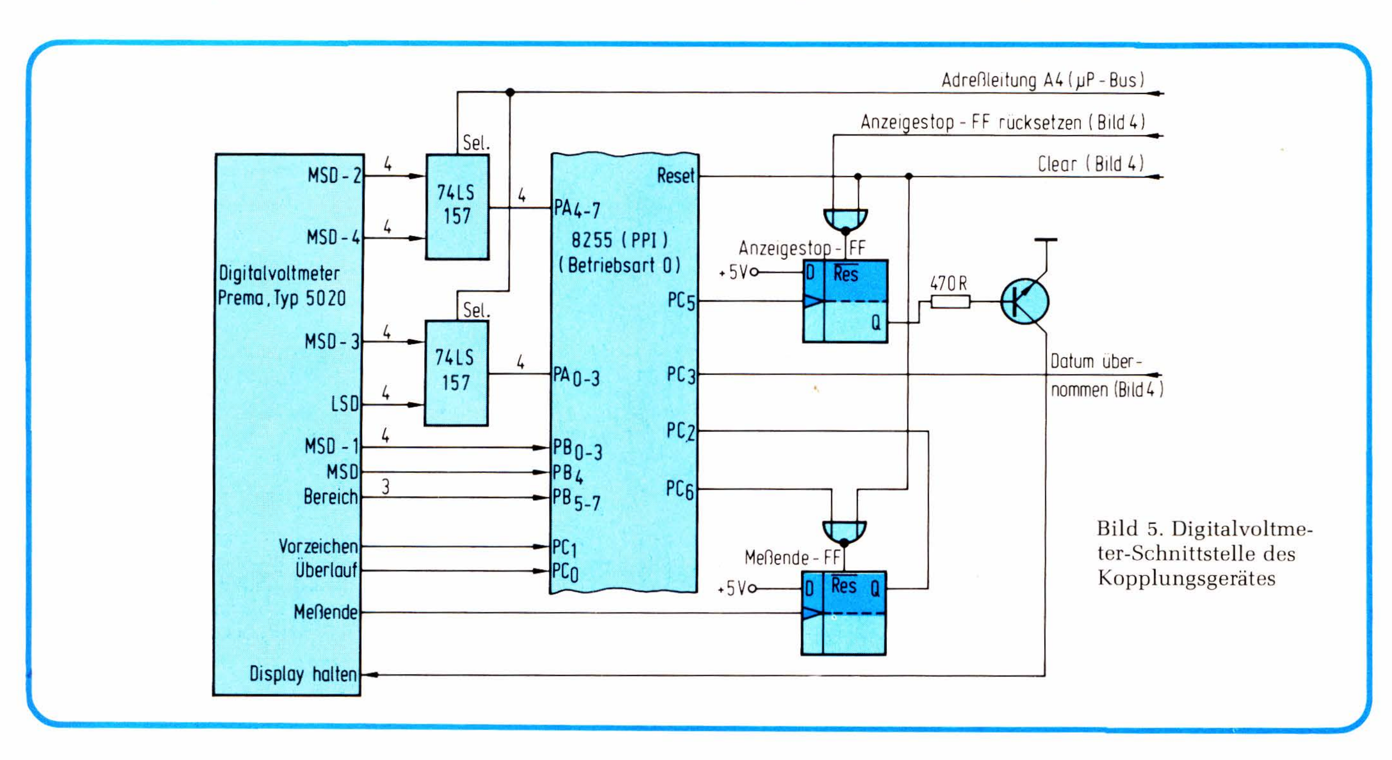


Bild 5. Digitalvoltmeter-Schnittstelle des Kopplungsgerätes

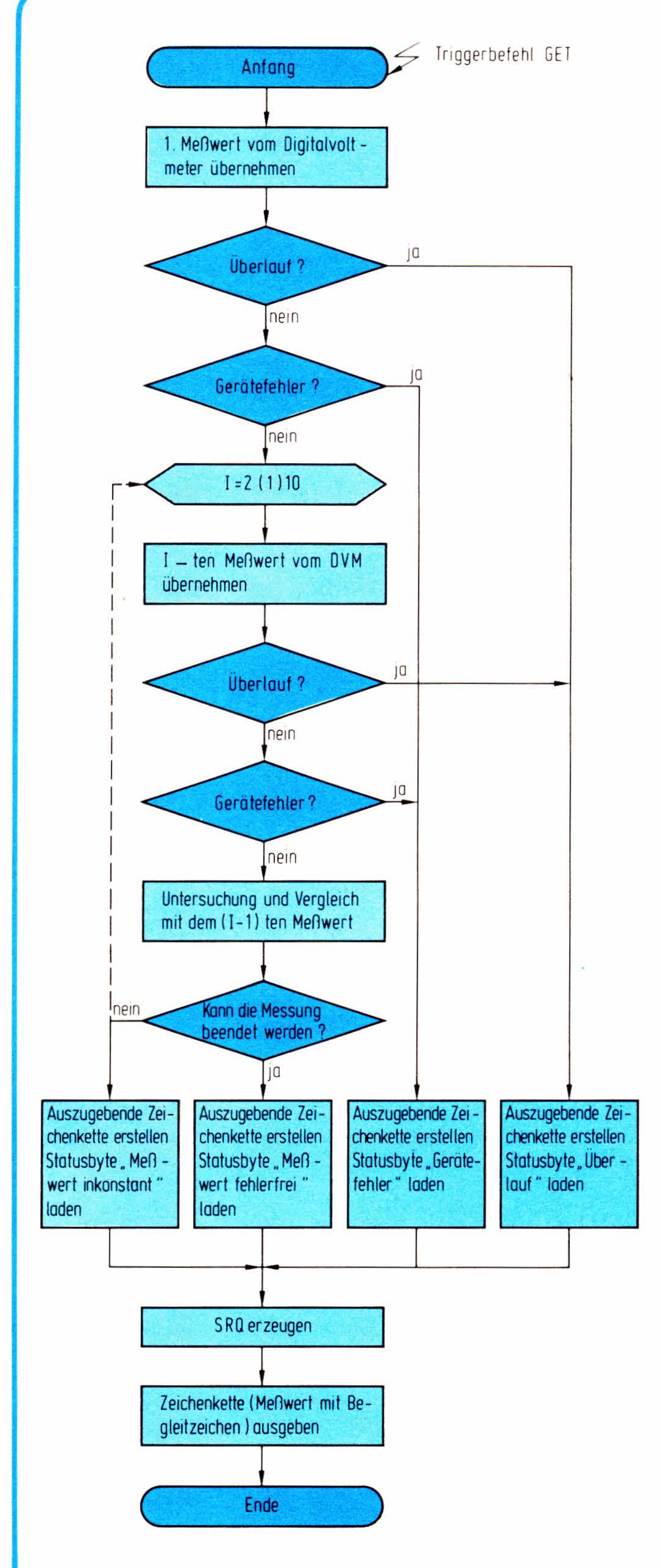


Bild 6.

a) Auslösung eines Gerätezyklus als Folge eines Triggerbefehls.

b) Überwachung des Meßvorganges

- Erkennen eines Überlaufs, wenn im größten Meßbereich (± 1000 V) der Anzeigeumfang des Digitalvoltmeters überschritten wird. Als Begleitzeichen und Meßwert wird die Zeichenkette 0+9,99999E+99 ausgegeben.
- Erkennen von Fehlern an der Schnittstelle zwischen Digitalvoltmeter und Kopplungsgerät (z. B. defekte Leitung). In diesem Fall wird der erste fehlerbehaftete Meßwert zur Übertragung angeboten. Ein Fehler im Meßbereich wird durch E+99 gekennzeichnet.
- Erkennen von Bedingungen, die das Meßergebnis verfälschen. Das Meßergebnis ist falsch, wenn sich die Meßspannung zum Zeitpunkt eines Triggerbefehls so ändert, daß eine oder mehrere Bereichsumschaltungen stattfinden. Im µP-Programm wird auf die verschiedenen gerätespezifischen Reaktionen, die das Digitalvoltmeter PREMA 5020 im Automatikbereich zeigt, eingegangen:

I) Verfälschte Meßgröße, die auf asynchrone IEC-Bus-Triggerung gegenüber dem Meßanfang im Digitalvoltmeter zurückzuführen ist. Ferner zeigen Meßgeräte nach Umschalten in den endgültigen Meßbereich aperiodisches Einschwingverhalten, das die Meßgröße verfälscht.

II) Messung im zu kleinen Bereich bei Meßspannungserhöhung vor der Umschaltung in einen größeren Bereich. Hierbei meldet das Digitalvoltmeter Überlauf („unechter" Überlauf).

III) Messung im zu großen Bereich bei Meßspannungserhöhung vor der Umschaltung in einen kleineren Bereich (mehr als eine vorlaufende Null).

IV) Falschmessung bei Polaritätswechsel der Meßspannung: Das Digitalvoltmeter meldet den Spannungswert von 0 V.

Im praktischen Betrieb hat sich herausgestellt, daß es sinnvoll ist, eine maximal zulässige Meßspannungsänderung zu definieren:

V) Ein Meßergebnis wird verfälscht, wenn sich die Meßspannung um ≥ 1‰ des aktuellen Meßbereichs pro Umsetzungszeit des Digitalvoltmeters ändert. Durch Untersuchung der Meßwerte und Vergleich zweier aufeinanderfolgender Meßwerte entscheidet das µP-Programm, wann ein Meßwert mit entsprechendem Begleitzeichen zur Ausgabe gelangt. Die Meßzeit durch das Kopplungsgerät ist ein Vielfaches der Umsetzungszeit; dabei kann der Faktor z. B. bei 1 (Überlauf und Gerätefehler), 2 (konstante Meßspannung) und 10 im worst-case Fall liegen (denn 4x Bereichsumschalten, 3x unzulässige Meßspannungsänderung im Bereich und 2x Meßwert gleich Null sind erlaubt). Tabelle 1 zeigt als Beispiel, welche Meßwerte das Kopplungsgerät vom Digitalvoltmeter erhält, wenn sich die Meßspannung sprungartig von 0,1 V auf 5 V ändert. Beim Vergleich zwischen Meßwert 4 und 5 wird eine unzulässige Meßspannungsänderung (≥0,01 V) festgestellt, der Vergleich zwischen Meßwert 5 und 6 verläuft positiv; letzterer wird zur Ausgabe über den IEC-Bus verarbeitet. In diesem Beispiel beträgt die Meßzeit das 6fache der Umsetzungszeit.

c) Normierung des Meßwerts; Erzeugung eines Ausgabeformats, das den Vorschlägen zur Zahlendarstellung auf dem IEC-Bus entspricht. Ein Begleitzeichen (T-Feld) weist auf Fehler hin. Ein Teil der Fehler führt außerdem zu nicht plausiblen Zahlenwerten, die eindeutig erkannt werden können.

d) Absenden einer Bedienungsanforderung an das Steuergerät, sobald c) beendet wurde; gleichzeitig wird eines der 4 möglichen Statusbytes (Tabelle 2) im Statusbyte-Ausgaberegister bereitgestellt, aus dem es vom Steuergerät im Zuge einer seriellen Abfrage ausgelesen werden kann. (Die Darstellung der Zustandsinformation sowohl im Meßwert selbst als auch mit Hilfe von Statusbytes trägt den Anforderungen unterschiedlichster IEC-Bus-Systeme Rechnung).

e) Ausgabe von Begleitzeichen und Meßwert auf den IEC-Bus, sobald das Kopplungsgerät als Sprecher adressiert wurde. Das Kopplungsgerät sendet die bereitgestellte Zeichenkette auch dann, wenn das für den Meßwert erarbeitete Statusbyte vom Steuergerät nicht abgeholt wurde. (Wird das Kopplungsgerät vor Beendigung des Abschnitts d) zum Sprecher adressiert, so ist zu beachten, daß das IEC-Bus-System solange blockiert wird, bis der Gerätezyklus abgeschlossen ist)

eindeutige Zuordnung Triggerbefehl – Meßwert besteht.

Durch den „adressierten“ Befehl SDC („Selected Device Clear“) oder den Universalbefehl DCL („Device Clear“) wird der Oclr-Ausgang im Zustand DTAS aktiv. Das Signal Clear (Gerätefunktion des Kopplungsgeräts zurücksetzen) ist die logische ODER-Verknüpfung aus dclr,pon („Power on“) und dem Signal des Rücksetz-Tasters. Durch Clear wird die Ausgangsstellung des Kopplungsgeräts hergestellt, d. h. die CPU führt einen Halt-Befehl aus.

Die 3 Datenquellen „Datenkanal“, „Statuskanal“ und „Parallel Poll-Antwort“ werden über Tri-State-Treiber (74LS240 und 74LS241) zusammengeführt und auf Bustreiber-Empfängerbausteine vom Typ MC3441 geführt, die die IEC-Bus-Normenanforderungen erfüllen. Welche der 3 Datenquellen auf den IEC-Bus durchgeschaltet wird, bestimmen die Zustände TACS, SPAS und PPAS in eindeutiger Weise.

4 Gerätefunktionen

Der Mikrocomputer des Kopplungsgeräts paßt die Eigenschaften des Digitalvoltmeters an die IEC-Bus-Konventionen an. Seine Hauptaufgaben in zeitlicher Abfolge sind in *Bild 6* dargestellt; *Tabelle 1* zeigt ein Meßbeispiel (Spannungsänderung von 0,1 auf 5 V).

5 Ausgabeformat und Übertragungsgeschwindigkeit

Die ASCII-Zeichenkette (vgl. DIN 66 003), die nach jedem Triggerbefehl und anschließender Rückmeldung mittels SRQ vom Kopplungsgerät angeboten wird, hat in allen Fällen die Länge 15 und entspricht dem Normentwurf für Meßdatenformate [6]; sie ist in *Tabelle 2* dargestellt. Zur Normierung des Meßwertes ist die einzige Vorkommastelle ungleich 0. Sie ist nur dann Null, wenn auch das Meßergebnis 0 V vorliegt, d. h. wenn die maximale Auflösung des Digitalvoltmeters unterschritten wird.

Die Übertragungsgeschwindigkeit auf den IEC-Bus beträgt 15 kByte/s und hängt ab von den Eigenschaften des Prozessors 8080. Die Übertragung der 15 Zeichen eines Meßwertes über den IEC-Bus dauert somit mindestens 1 ms.

6 Statusmeldungen

Das Kopplungsgerät kann zu jedem Zeitpunkt, auch während einer Meßwertausgabe, auf Anforderung ein Statusbyte senden. Von Bedeutung ist jedoch nur das Statusbyte, das nach einer aktiven Bedienungsanforderung durch das Steuergerät abgeholt werden sollte, bevor ein neues Triggersignal erfolgt. Damit besteht eine eindeutige Zuordnung zwischen Erteilung eines Geräteauftrages (Messung), dessen Beendigung und dem dazugehörigen Statusbyte (*Tabelle 3*).

Weiterhin besitzt das Kopplungsgerät die Parallel-Poll-Funktion, mit deren Hilfe das Steuergerät erkennen kann, welche Geräte im IEC-Bus-Verbund eine Bedienungsanforderung abgesetzt haben. In [7] sind „Bedienungsruf“ und „Parallelabfrage“ eingehend beschrieben.

7 Besondere Eigenschaften des Gerätes

Eine wichtige Eigenschaft des Kopplungsgeräts ist, daß es nach Erteilung eines Geräteauftrages durch das Steuergerät unabhängig vom IEC-Bus arbeitet. Es

Tabelle 2. Ausgabeformat

Position	Feld	ASCII-Zeichen	Erläuterungen, Bedingungen bei Alternativen
1	T	„ “ (Space) „V“ „O“ „F“	Meßwert fehlerfrei Eingangsspannung inkonstant Eingangsspannung > ±1000 V Gerätefehler
2	U	„+“ „–“	Vorzeichen der Mantisse
3	V	„1“...„9“ „0“	Einzige Vorkommastelle der Mantisse (normierte Darstellung) Meßwert ist Null. Ergibt sich wenn Eingangsspannung < 1 μV
4	V	„.“	Dezimalpunkt
5 . . 9	V	„0“...„9“	1...5. Nachkommastelle der Mantisse
10	W	„E“	Exponentenzeichen
11	W	„+“ „–“	Vorzeichen des Exponenten
12	W	„0“...„9“	Zehnerstelle des Exponenten
13	W	„0“...„9“	Einerstelle des Exponenten
14	Z	„CR“	ASCII-Steuerzeichen „Carriage return“
15	Z	„LF“	ASCII-Steuerzeichen „Linefeed“

Tabelle 1. Verhalten des Digitalvoltmeters im Automatik-Meßbereich bei Meßspannungssprung von +0,1 auf +5 V

Meßwert-Nr.	Meßwert	Ursache
1	+1.60113E-01	asynchrone Triggerung
2	Überlauf	Umschaltung von dem 0,1-V- in den 1-V-Meßbereich
3	Überlauf	Umschaltung von dem 1-V- in den 10-V-Meßbereich
4	+0.50214E+00	aperiodisches Einschwingen
5	+0.50000E+00	1. fehlerfreier Meßwert
6	+0.50000E+00	2. fehlerfreier Meßwert

Tabelle 3. Statusbytes

IEC-Bus-Datenleitungen DIO 8	7	6	5	4	3	2	DIO1	ASCII-Zeichen	Bedeutung
0	1	0	0	0	0	0	0	@	Meßwert fehlerfrei
0	1	0	0	0	0	0	1	A	Überlauf
0	1	0	0	0	0	1	0	B	Meßwert inkonstant
0	1	0	0	0	0	1	1	C	Gerätefehler

meldet sich dann zurück, wenn der Geräteauftrag erfüllt ist, d. h. wenn das Kopplungsgerät als Sprecher adressiert werden kann, um über den IEC-Bus einen Meßwert zu senden. Dies ist besonders wichtig, da insbesondere hochauflösende Präzisionsmeßgeräte eine große Umsetzzeit haben, die bis zu einigen Sekunden betragen kann. Während des Gerätezyklus, dessen Dauer von der Umsetzungszeit abhängt, kann das Steuergerät andere Aufgaben durchführen.

Die gerätespezifischen Eigenschaften des Digitalvoltmeters werden durch die Software des Kopplungsgeräts berücksichtigt. Ein modularer Aufbau des Programms erleichtert notwendige Änderungen, wenn ein Digitalvoltmeter anderen Typs angeschlossen werden soll. Grundsätzlich eignen sich alle Meßgeräte zur Kopplung, die die in Bild 1 beschriebenen parallel anstehenden Signale über einen Stecker nach außen führen.

Literatur

[1] Entwurf DIN IEC 625, Teil 1, Elektrische Meßtechnik, „Byteserielles bitparalleles Schnittstellensystem für programmierbare Meßgeräte", Beuth-Verlag GmbH, Berlin-Köln 1976.

[2] Bedienungsanleitung Digitalvoltmeter Typ 5020, Unterlagen der Fa. Prema, Mainz.

[3] *Dietsch, H.*: Microprocessors in IEC-Bus-compatible units, in: Microprocessors in Automation and Communications, IERE Conference Proceedings, London, 1978.

[4] *Exalto, J. P.*: The HEF 4738V IEC Bus interface circuit, Technische Informationen für die Industrie, Fa. Valvo, Hamburg.

[5] Datenblätter und Applikationen zur 8080-μC-Familie der Firmen Siemens und Intel.

[6] DIN IEC 66 CO 31, erscheint Ende 1979.

[7] *Dietsch, H.*: Prinzip und Praxis der IEC-Bus-Funktionen „Bedienungsruf" und „Parallelabfrage", ELEKTRONIK 1978, H. 12, S. 89...96.



Ing. (grad.) Georg Unger ist in Wolfsburg geboren. Nach dem Studium der Nachrichtentechnik an der Fachhochschule Regensburg war er $2^1/_2$ Jahre als Entwicklungsingenieur für Unfallmelde- und Notrufanlagen bei der TEKADE in Nürnberg tätig. Seit 1975 ist er am Lehrstuhl für Technische Elektronik – Abteilung Informatik Forschungsgruppe 7 – der Universität Erlangen-Nürnberg beschäftigt. Sein Hauptarbeitsgebiet sind das Entwickeln von IEC-Bus-Systemen und der Einsatz von Mikroprozessoren.
Bevorzugtes Hobby ist die Aquaristik.
Diensttelefon: (0 91 31) 85 78 68, Privattelefon: (0 91 26) 13 42
ELEKTRONIK-Leser seit 1972

Übersicht: IEC-Bus-Interface-ICs

Der folgende Übersichtsbeitrag stellt einen Auszug aus einer an der Universität Bremen durchgeführten Arbeit dar. Er bringt einen Einblick in die zur Zeit auf dem Markt befindlichen Bus-ICs und -Baugruppen.

1 Übersicht

In der *Tabelle 1* sind die Interface-ICs mit den wichtigsten Daten zur Vorauswahl angeführt (Datenblätter der Hersteller sind natürlich zur genaueren Information unabdingbar).

Es sei noch bemerkt, daß die Firma Motorola einen Baustein mit Controllerfunktion plant. Dieser soll dem Intel 8292 ähnlich sein. Weiterhin plant National Semiconductor einen IEC-Bus-Baustein, der außer den Talker-/Listener- und Controllerfunktionen auch die Treiberstufen enthält.

In Tabelle 1 sind zwei Bausteine aufgeführt, die von ihrer Konzeption her für unintelligente Geräte entworfen sind. Dies sind der Valvo-Baustein HEF 4738V und der Fairchild-Baustein 96LS488. Mit diesen Bausteinen kann auch eine Schnittstelle für einen Rechner realisiert werden. Diese Realisierung erfordert jedoch einen erhöhten Aufwand.

Eine weitere Untersuchung befaßte sich mit kompletten Bus-Baugruppen, die zumeist als Einschübe oder Platinen für Komplettgeräte (Rechner) der entsprechenden Firmen hergestellt werden (Nachrüstung). Eine Übersicht über die untersuchten Baugruppen zeigt die *Tabelle 2*. Es fanden nur solche Einschübe Beachtung, für die an der Universität Bremen entsprechende Geräte vorhanden waren; dennoch soll das Ergebnis hier nicht vorenthalten werden.

2 Allgemeines

Der Aufbau einer Schnittstelle ist vor allem davon abhängig, ob die Schnittstelle mit einem intelligenten oder unintelligenten Gerät zusammenarbeiten soll. Schnittstellen, die mit einem intelligenten Gerät zusammenarbeiten, haben an der Geräteseite den entsprechenden Datenbus des Rechners zur Kommunikation. Bei Rechneranwendungen sollte außerdem ein Interruptkonzept zwecks schneller Verarbeitung verwendet werden. Bei einem unintelligenten Gerät wird dagegen die Dateneingabe und Datenausgabe getrennt. Die Kommunikation findet hier über einzelne Leitungen, die eine bestimmte Funktion haben, statt.

Bei der Anwendung spielt natürlich auch die Frage eine Rolle, welche Funktionen der Norm realisiert und ob diese voll oder nur zum Teil implementiert werden

Tabelle 1. Untersuchte IEC-Bus-Interface-ICs

Hersteller Typ	Techno-logie	Gehäuse	Preis (ungef.)	Anwendung für	Funktion [1]	zusätzl. Bauteile nötig	Übertr.-rate
Valvo HEF 4738 V	LOCMOS	40pol. DIL	120.–	unintell. Gerät	T/L	4 Treiber, 1 Decod., 2 Multipl., 2 Schieber.	bis 200 kByte/s
Motorola MC 68488	NMOS	40pol. DIL	60.–	Gerät mit 8-Bit-μP	T/L	4 Treiber Taktgenerator	ca. 120 kByte/s
Fairchild 96 LS 488	LSTTL	48pol. DIL	100.– bis 200.–	unintell. Gerät	T/L	3 Multiplexer 1 Treiber	250 kByte/s 1 MByte/s
Hewl.-Pack. PHI	CMOS/ SOS	40pol. DIL	– [2]	Gerät mit 8/16-Bit-μP	T/L	keine Angaben	200 kByte/s
Intel GPIB 8291	NMOS	40pol. DIL	120.–	Gerät mit 8/16-Bit-μP	T/L	2 Treiber	keine gen. Angaben
Intel GPIB 8292	NMOS	40pol. DIL	?	siehe 8291	C	GPIB 8291	siehe 8291
Texas Instr. TMS 9914	NMOS	40pol. DIL	96.–	Gerät mit 8/16-Bit-μP	C/T/L	Treiber (75160/1	noch keine Ang.

[1]) T = Talker, L = Listener, C = Controller; [2]) Anwendung nur firmenintern

sollen. Bei den IEC-Bus-Bausteinen ist die Konzeption im allgemeinen so, daß der Baustein eine Funktion vollständig enthält, wenn diese überhaupt verwirklicht wurde. Auch wurden bei den Bausteinen meist alle Funktionen bis auf die Controllerfunktion verwirklicht. Die Schnittstellenbaugruppen enthalten dagegen nicht alle Funktionen und diese auch oft nur in Teilausstattungen. Hier liegt dann z. B. die Konzeption vor, daß das Gerät der einzige Controller im IEC-Bus-System ist. Eine Verwendung bestimmter Funktionen ist aber in vielen Anwendungsfällen völlig ausreichend.

Bei der Konzeption einer IEC-Bus-Schnittstelle muß ebenfalls die maximale Datenübertragungsrate festgelegt werden. Es muß klar sein, ob die Schnittstelle mit der maximalen Datenübertragungsrate von 1 MByte/s und Tri-State-Treibern oder mit 250 kByte/s und Open-Kollektor-Treibern arbeiten soll. Die hier angegebenen Bus-Baugruppen arbeiten alle mit der geringeren Datenübertragungsrate; die Bus-ICs ermöglichen dagegen oft beide Werte.

Zusammengefaßt zum Schluß noch einmal die Realisierungsmöglichkeiten:

- Talker/Listener Realisierungen in sequentieller Logik für unintelligente Geräteanwendungen (HEF 4738V, 96LS488)
- Talker/Listener Realisierungen in sequentieller Logik für intelligente Geräteanwendungen (MC68488, GPIB 8291, PHI)
- Talker/Listener/Controller Realisierungen in sequentieller Logik für intelligente Geräteanwendungen (IEC-11A, IBV-11A, TMS 9914, HP 59405)
- Schnittstellen mit eigenem Mikroprozessor (HP 98034, GPIB 8292).

Diese Arbeit wurde vom Bundesministerium für Forschung und Technologie im Rahmen eines DIN-Projektes unterstützt und unter der wissenschaftlichen Betreuung von Prof. Dr.-Ing. D. Popovic (Fachgebiet „Prozeßrechentechnik" der Universität Bremen) durchgeführt.

Tabelle 2. Untersuchte IEC-Bus-Baugruppen

Typ	Hersteller	Funktion [1]	zugehöriges Gerät	Preis	Übertr.-rate	Bemerkungen
HP 59405A	HP	C/T/L	HP 9830	ca. 1800.–	2 kByte/s	eigene Tests durchgeführt
HP 98034A	HP	C/T/L	HP 9845	k. Ang.	45 kByte/s	Gerät stand nicht zur Verfügung
IBV-11A	DEC	C/T/L	PDP 11/03	2318.–	40 kByte/s	eigene Tests durchgeführt
IEC-11A	DEC	C/T/L	PDP 11/45	5160.–	k. Ang.	Schnittstelle noch nicht geliefert

[1]) T = Talker, L = Listener, C = Controller

Rolf-Dieter Klein

Der PET und der IEC-Bus

Zur Kommunikation mit der Außenwelt besitzt der PET neben einem Parallel-Port auch einen IEC-Bus-Anschluß, wobei die Anschlußbelegung der IEEE-488-Norm entspricht. Der IEC-Bus gestattet es, eine Vielzahl von Meßgeräten, Plottern und Druckern sowie Floppy-Laufwerke mit dem PET zu verbinden. Leider entsprechen hierbei oft nicht alle Signale exakt der Norm, so daß sich bei manchen IEC-Bus-Geräten Schwierigkeiten ergeben.

1 Prinzipieller Ablauf des Handshakes beim PET-IEC-Bus

Das *Bild* zeigt das dazugehörige Impulsdiagramm.

a im Ruhezustand ist die NRFD-Leitung auf *not ready*, DAV auf *data not valid* und NDAC auf *data not accepted*. Wird nun zum Beispiel das Peripheriegerät fertig, wenn es als Listener arbeitet und der PET als Talker, so zeigt es dies durch „Hochnehmen" der NRFD-Leitung auf *ready for data* an. Ist der PET gerade Listener und das Gerät Talker (z. B. nach einem INPUT-Befehl), so

b nimmt der PET die Leitung NRFD auf *ready for data* und die Daten müssen spätestens nach 64 ms mit DAV als gültig angezeigt werden, sonst erscheint eine Fehlermeldung im PET-Status.

c DAV geht auf *data valid* und zeigt damit an, daß die Daten auf dem Datenbus verwendet werden können.

d Vom Listener muß nun als erstes die NRFD-Leitung wieder auf *not ready for data* genommen werden und dann

e die Leitung NDAC auf *data accepted*, um dem Talker anzuzeigen, daß die Daten aufgenommen wurden.

f Dieser reagiert dann durch DAV auf *data not valid* und nimmt die Daten weg vom Datenbus, sobald

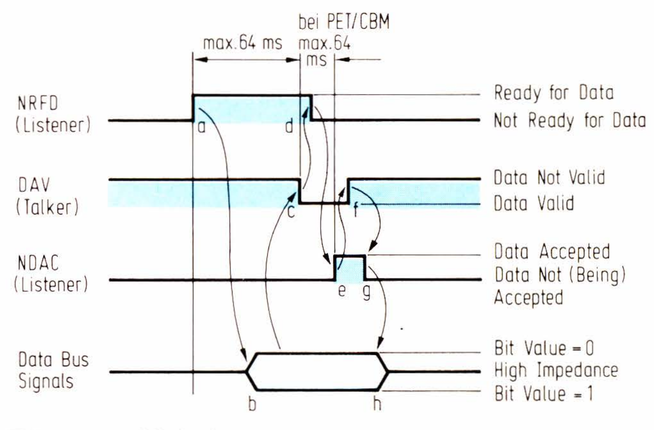
g der Listener NDAC wieder auf data not accepted gebracht hat.

Eine Zeitüberwachung von seiten des PET liegt auch vor, wenn er als Talker DAV auf *data valid* schaltet und NDAC abwartet; sie muß in mindestens 64 ms auf *data accepted* gehen.

Diese Zeitüberwachung macht insbesondere bei langsamen Speichern Schwierigkeiten, die über den IEC-Bus angeschlossen werden. Wird ein Richtungswechsel beim PET durchgeführt, also von Talker auf Listener (ggf. durch einen INPUT- oder GET-Befehl), so müssen die Daten spätestens nach 64 ms anstehen, da es nicht möglich ist, das Handshake bis zur Aufnahme der Daten vorher anzuhalten. Ist einmal das erste Byte gegeben, so kann das Handshake zum Beispiel durch Belassen des DAV im Zustand *data valid* beliebig lange angehalten werden. Probleme gibt es also nur beim ersten Byte. Gerade bei Floppys oder Platten muß dieses Byte aber unter Umständen erst geholt werden, und diese Zeit ist viel größer als 64 ms. Bei dem INPUT-Befehl ergeben sich keine Schwierigkeiten, da dem PET beliebig viele „Blanks" geschickt werden können, die er dann ignoriert; nur bei GET ist dies nicht möglich, und Abhilfe läßt sich nur durch ein „vorbeugendes" Holen des Datensatzes vor der Ausführung des GET-Befehls schaffen.

2 Einzelne IEC-Management-Belegungen

ATN Attention wird vom PET auf „L" gesetzt, wenn er Kommandos über den Bus schickt. Wenn ATN



IEC-Bus-Handshake beim PET

auf „L“ ist, sind immer nur Kommandos oder Peripherieadressen auf dem Datenbus.
Ist ATN auf einem „H“-Pegel, so können nur zuvor adressierte Geräte auf dem Bus Daten austauschen.

EOI End or Identify. Wird das letzte Byte eines Datensatzes übertragen, so kann EOI auf „L“ gesetzt werden. Der PET setzt EOI immer beim letzten Byte auf „L“, um das Ende des Datenstromes anzuzeigen. Ein angeschlossenes Gerät muß EOI setzen, wenn es solche Daten zum PET überträgt (deren letztes Zeichen nicht CR ist), um dem PET das Ende der Übertragung anzuzeigen.

IFC Interface clear. Nach einem RESET oder beim Einschalten des PET wird dieses Signal für ungefähr 100 ms auf „L“ gelegt.

SRQ Service request. Wird vom BASIC nicht benützt, ist aber durch POKE-Befehle setzbar.

REN remote enable. REN wird vom PET permanent auf „L“ gelegt.

3 Statusmeldungen des PET

Über die Variable ST, die als „Nur-Lese-Speicher“ fungiert, kann vom BASIC aus der Status des IEC-Transfers abgelesen werden.

1 bedeutet Zeitüberlauf bei Listener; es wurde nach 64 ms keine Reaktion gezeigt

2 bedeutet Zeitüberlauf bei Talker. DAV zeigt nach 64 ms seit *ready for data* keine Antwort.

64 EOI ging beim letzten Datenbyte auf „L“.

−128 DEVICE NOT PRESENT. Hier verläßt das BASIC die Ausführungsphase und geht in den BASIC-Kommando-Zustand.

Tabelle der IEC-Bus-Anschlußbelegung beim PET

PET-Kontakt-Nr.	IEEE-Bezeichnung	Nr. (durchlaufend)
1	DIO1	1
2	DIO2	2
3	DIO3	3
4	DIO4	4
5	EOI	5
6	DAV	6
7	NRFD	7
8	NDAC	8
9	IFC	9
10	SRQ	10
11	ATN	11
12	SHIELD (Abschirmg.)	12
A	DIO5	13
B	DIO6	14
C	DIO7	15
D	DIO8	16
E	REN	17
F	GND	18
H	GND	19
J	GND	20
K	GND	21
L	GND	22
M	GND	23
N	LOGIC GND	24

4 Ablaufbeispiele für verschiedene BASIC-Befehle

OPEN 1,8,4,"INPUT"
wird an den IEC-Bus wie folgt übermittelt:

	ATN	28	Device-Adresse 8, Listener
	ATN	F4	Secondary Adresse 4, Open Befehl da Fx
		I	
		N	
		P	
		U	
EOI		T	
	ATN	3F	Unlisten-Zeichen

PRINT #1, "TEST";

	ATN	28	Device-Adresse 8, Listener
	ATN	64	Sec.Adr.4,6xbedeutet„Print-Befehl“
		T	
		E	
		S	
EOI		T	
	ATN	3F	Unlisten

INPUT #1, A$ Eingabe eines Strings

	ATN	48	Device-Adresse 8, Talker
	ATN	64	Input-Befehl Sec.Adr.4
		41	Eingabe erfolgt von Gerät
		42	
EOI		43	letzte Eingabe durch Gerät
	ATN	5F	Untalk von PET ausgesendet

A$ enthält nun den String ABC

CLOSE 1 Schließen der File

ATN	28	Device Adr.8
ATN	E4	Close von Sec.Adr.4
ATN	3F	Unlisten

Bei Ausführung des LOAD-Befehls wird der Sekundärkanal 0 und beim SAVE-Befehl der Sekundärkanal 1 verwendet.

Dabei wird beim LOAD "xxxx",devadr
zunächst ein OPEN 1,devadr,0, "xxxx" durchgeführt,
dann
ein INPUT#1, to memory
und schließlich ein CLOSE 1.

Bei SAVE ist es entsprechend, nur daß die Sekundäradr. 1 verwendet wird und die Daten vom Speicher zum Gerät hin übertragen werden.

Literatur

„PET Communication with the outside world“. Ergänzung zum PET-Benutzerhandbuch von Commodore.

Automatischer Hydraulik-Ventilprüfstand mit IEC-Bus

Eine interessante Einrichtung ist das von der Firma Meilhaus Electronic (München) zusammengestellte Meßwerterfassungssystem für einen Hydraulik-Ventilprüfstand.

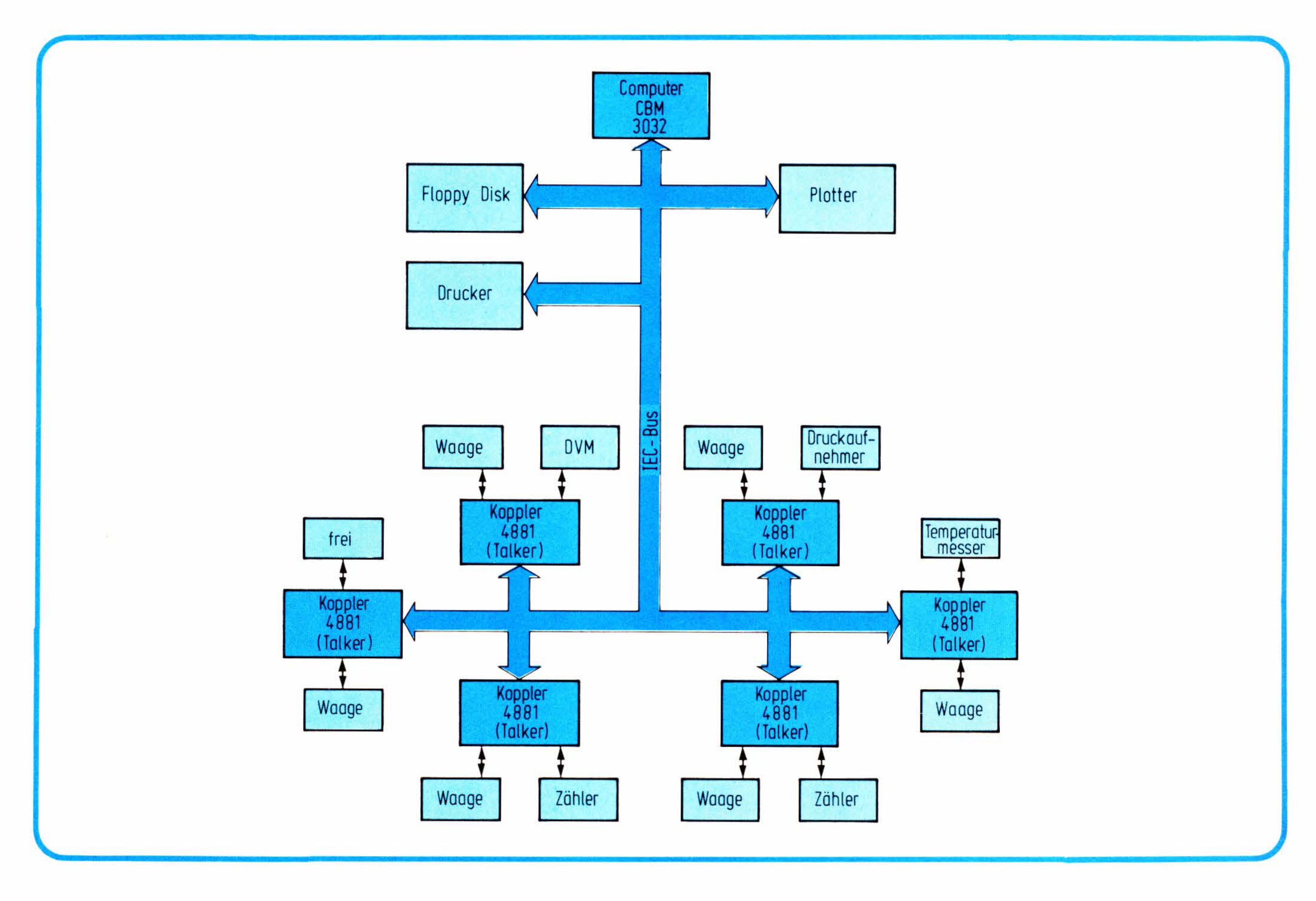
Die Anlage hat die Aufgabe, vorhandene und mit BCD-Schnittstellen versehene Meßgeräte an einen IEC-Bus-fähigen Rechner, hier den Commodore CBM-3032, zu adaptieren. (Grundsätzlich ist es natürlich möglich, jeden beliebigen IEC-Bus-fähigen Rechner zu verwenden.) Der Anschluß der Meßgeräte wurde mit BCD/IEC-Bus-Kopplern der Fa. ICS Electronics Corp. durchgeführt (*Bild*). Jeder dieser Koppler besitzt einen 10-Dekaden-BCD-Eingang, und unter der Voraussetzung, daß zwei kombinierte Meßgeräte nicht mehr als 10 Dekaden liefern, ist es somit möglich, je zwei Meßgeräte an einen Koppler zu schalten. Neben sechs Waagen zur Messung der abgegebenen Flüssigkeitsmenge auf sechs Zylinder werden zwei Zähler zur Ermittlung der Einspritzdauer, sowie Temperaturmeßgerät, Spannungsmesser und Druckaufnehmer für die Aufnahme der Rahmenbedingungen benutzt.

Nach Herstellen aller elektrischen Anschlüsse und Aufbau des hydraulischen Kreises sowie Einlesen der entsprechenden Software von der Floppy-Disk kann mit der Messung begonnen werden.

Während einer Meßreihe liest der Rechner alle notwendigen Meßwerte über die Koppler von den Meßgeräten und speichert diese zwischen, um sie im Anschluß auf eine IEC-Bus-kompatible Floppy-Disk aufzuzeichnen. Nach der Weiterverarbeitung dienen ein buskompatibler Drucker und Plotter dazu, die Messungen zu dokumentieren und die spezifischen Ventilkennlinien für nachfolgende Auswertungen aufzuzeichnen.

Mit dem Einsatz dieser Anlage wird eine wesentliche Zeitersparnis beim Erstellen von Ventilkennlinien erreicht. Diese Zeitersparnis soll – je nach gewünschter Genauigkeit – bis über 80 % betragen. Dieses Beispiel unterstreicht also die universelle Einsetzbarkeit des IEC-Bus-Systems in Meßanordnungen, die bisher manuell bedient werden mußten, vor allem auch unter Beibehaltung bzw. Erhöhung der Meßgenauigkeit.

Jürgen Geltinger





Elektronik
3
Fachzeitschrift für angewandte Elektronik und Datentechnik
8. Februar 1979
4.20 DM
36.–öS, 4.50 sfr
REPORT: Leistungs-Thyristoren
Meßdatenerfassungsanlage mit IEC-Bus
Im Westen viel Neues – Programm für µC-gesteuerte Längenmessung – Monostabiler Multivibrator ohne Kapazität

Leistung verbindet

Die FUNKSCHAU, Fachzeitschrift für Rundfunk- und Fernsehtechnik, Elektronik, Hi-Fi, Audiovision, Mikrocomputer, ist die führende deutschsprachige Zeitschrift in ihrem Bereich.

Einen Schwerpunkt bildet in jedem Heft der sehr umfangreiche aktuelle Teil einschließlich funkschau-express: mindestens zehn Druckseiten frische Nachrichten und „heiße" Kurzberichte aus der weiten Welt der Elektronik. Regelmäßig werden auch die wirtschaftlichen Trends der Branche aufgezeigt.

Ständige Rubriken sind: 6 Seiten Hi-Fi-Technik, 8 Seiten Mikrocomputer mit vielen extra erstellten Rechnerprogrammen, der beliebte und praxiserprobte Selbstbauteil praxis & hobby (mindestens 6 Seiten), die in Abständen erscheinenden Funktechnischen Arbeitsblätter und der ständige Neuheitenmarkt mit Produktbesprechungen.

Ein wesentlicher Teil der FUNKSCHAU konzentriert sich außerdem auf Forschungsergebnisse, Digital- und Meßtechnik, Beschreibung von Bauelementen sowie auf das Verfolgen der Fortschritte der Halbleitertechnik.

Hervorzuheben ist auch die ausführliche Berichterstattung über die einschlägigen Messen und Ausstellungen – im In- und Ausland. Die FUNKSCHAU garantiert, daß Sie nichts Wissenswertes versäumen. Diese Leistung verbindet uns mit immer mehr Fachleuten, die etwas leisten wollen. Mehrere Sonderthemenhefte, über das Jahr verteilt, befassen sich mit interessanten Themen (z.B. Heim-Videotechnik) in aller Ausführlichkeit. Das Gespräch mit prominenten Persönlichkeiten der Branche wird besonders gepflegt, entweder als FUNKSCHAU-Gespräch oder als Besuch in der Redaktion. Weitere Schwerpunkte sind die Satellitentechnik, die audiovisuellen Techniken und die Belange der Funkamateure und der CB-Freunde.

Eine gute Fachzeitschrift wie die FUNKSCHAU hat naturgemäß einen umfangreichen Anzeigenteil. Handel und Industrie stellen ihre Produkte vor, der Stellenmarkt ist beachtlich und die zahllosen Kleinanzeigen beweisen die enge Verbindung zwischen den Lesern und „ihrer FUNKSCHAU".

Die FUNKSCHAU erscheint 25mal im Jahr alle 14 Tage am Freitag. Das Einzelheft kostet DM 4.–, das Jahresabonnement DM 78.– im Inland, DM 88.– im Ausland (ab 1. 1. 81 DM 90.– bzw. DM 96.–).

Funkschau

die zuverlässige Verbindung zum Erfolg

Sie erhalten sie an jeder größeren Zeitschriftenverkaufsstelle zum Preis von DM 4.–. Ein Abonnement können Sie mit der Karte vor der hinteren Umschlagseite bestellen.

